

Erstellung einer Benutzeroberfläche zur Exploration und Analyse großer Datenmengen eines Ultraschall-Computertomographen

DIPLOMARBEIT

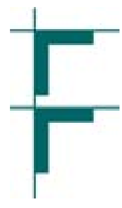
für die Prüfung zum
Diplom-Informatiker (Berufsakademie)

der Fachrichtung Angewandte Informatik
an der Berufsakademie Mannheim

von
Torsten Hopp

September 2006

Bearbeitungszeitraum	3 Monate
Kurs	TAI03AIM
Ausbildungsfirma	Forschungszentrum Karlsruhe GmbH Karlsruhe
Gutachter der Ausbildungsfirma	Dipl.-Inform. Gregor Schwarzenberg
Gutachter der Studienakademie	Jürgen Schultheis



Forschungszentrum Karlsruhe
Institut für Prozessdatenverarbeitung und Elektronik

Erstellung einer Benutzeroberfläche zur Exploration und Analyse großer Datenmengen eines Ultraschall-Computertomographen

Torsten Hopp

22. September 2006

Ehrenwörtliche Erklärung

Karlsruhe, 22. September 2006

Hiermit versichere ich, dass ich die vorliegende Diplomarbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

Eigenhändige Unterschrift des Verfassers

Danksagung

Ein besonderer Dank gilt meinem Erstbetreuer Gregor Schwarzenberg für die stets hilfreiche Unterstützung, die unzähligen Erläuterungen bei MATLAB-Problemen und das Aufzeigen von Verbesserungsmöglichkeiten für die Diplomarbeit - vielen Dank für eine vorbildliche Betreuung!

Des Weiteren möchte ich mich bei Herrn Jürgen Schultheis für die Übernahme der Zweitbetreuung sowie die zahlreichen Tipps bezüglich der formalen Aspekte der Diplomarbeit bedanken.

Außerdem danke ich allen Mitarbeitern der USCT-Softwaregruppe, insbesondere Nicole Ruiter und Michael Zapf für die Einbringung vieler Ideen und die gute Zusammenarbeit.

Ganz besonders bedanken möchte ich mich bei meiner Familie und meinen Freunden für das entgegengebrachte Vertrauen und die Unterstützung während des gesamten BA-Studiums.

Exposé

Am Institut für Prozessdatenverarbeitung und Elektronik des Forschungszentrums Karlsruhe wird ein neuartiges bildgebendes Verfahren zur Erzeugung von dreidimensionalen Bildern entwickelt, welches zur Brustkrebsfrüherkennung eingesetzt werden soll. Die so genannte Ultraschall-Computertomographie (USCT) basiert auf einer Vielzahl von Ultraschallwandlern, die bei einer Untersuchung die weibliche Brust vollständig umschließen.

Aufgrund der großen Anzahl von Ultraschallwandlern entstehen bei jeder Aufnahme riesige Datenmengen von etwa 20 Gigabyte. In der vorliegenden Diplomarbeit wird die Entwicklung einer Software beschrieben, die es den wissenschaftlichen Mitarbeitern der USCT-Arbeitsgruppe ermöglicht, die Daten zu durchsuchen, zu visualisieren und zu analysieren. Diese Software soll die Möglichkeit bieten, Zusammenhänge zwischen den aufgenommenen Daten und den rekonstruierten 3D-Bildern schneller zu erfassen und damit die Optimierung des derzeitigen USCT-Prototypen voranzutreiben.

Zunächst wird in der vorliegenden Diplomarbeit auf die Grundlagen der Ultraschall-Computertomographie und der Programmiersprache MATLAB eingegangen, bevor ein kurzer Überblick über die Prinzipien des guten Benutzeroberflächen-Designs gegeben wird. Darauf aufbauend wird das zugrunde liegende Konzept der Software erläutert. Anschließend geht die Diplomarbeit detailliert auf die Umsetzung der Anforderungen ein.

Zusammenfassung

Am Institut für Prozessdatenverarbeitung und Elektronik des Forschungszentrums Karlsruhe wird mit der 3D Ultraschall-Computertomographie ein neuartiges bildgebendes Verfahren zur besseren Brustkrebsfrüherkennung entwickelt. Es basiert auf ca. 2000 Ultraschallwandlern, die in Schichten ringförmig auf einem mit dem Koppelmedium Wasser gefüllten Zylinder angeordnet sind. Bei einer Aufnahme mit dem Ultraschall-Computertomographen (USCT) entstehen sehr große Datenmengen von ca. 20 GB, die als Rohdaten abgespeichert werden. Aus diesen Daten wird anschließend ein dreidimensionales Bild rekonstruiert.

In der aktuellen Entwicklungsphase des Prototypen ist es für die wissenschaftlichen Mitarbeiter der USCT-Arbeitsgruppe wichtig, möglichst einfach einen guten Überblick über den komplexen Datensatz und eine ganzheitliche Sicht auf das gesamte System zu erhalten. Mit der existierenden Software sind jedoch nur Sichten auf Teilbereiche möglich, weshalb Analysen oft händisch und unter erhöhtem Aufwand durchgeführt werden müssen. Um eine einfachere Exploration, Analyse und Visualisierung der Daten zu ermöglichen, wird im Rahmen der vorliegenden Diplomarbeit eine mit „DiagTool“ betitelte Software mit grafischer Benutzeroberfläche erstellt, die die drei Sichtweisen auf die Ultraschall-Computertomographie, d.h. Rohdaten, rekonstruierte Bilder und der Zusammenhang mit dem Entstehungsort im physikalischen USCT vereint.

Die in der Programmiersprache MATLAB implementierte Software besteht aus einer dreiteiligen Benutzeroberfläche, deren Teilbereiche den Sichtweisen auf den USCT entsprechen. Diese können sowohl isoliert als auch miteinander verknüpft gesehen werden. Trotz des großen Funktionsumfangs wurden die Oberflächen durch die Verwendung von Menüstrukturen übersichtlich gehalten. Der Benutzer erhält vielfältige Interaktionsmöglichkeiten für die Bedienung der Software, wie z.B. ein interaktives, dynamisch generiertes 3D-Modell. Über eine programmierbare linke Maustaste ist eine starke Vereinfachung und Beschleunigung von Selektionen darin möglich. Ein neuartiges Plugin-Konzept gewährleistet die beliebige Erweiterbarkeit der Software mit neuer Funktionalität.

Die Software kann es ermöglichen, Zusammenhänge für die USCT-Arbeitsgruppe erstmalig in einer integrierten Benutzeroberfläche und damit sehr viel schneller zu erschließen.

Abstract

At the Institute for Data Processing and Electronics at Forschungszentrum Karlsruhe, a new medical imaging system for early diagnosis of breast cancer is currently delevoped. The so-called 3D Ultrasound-Computer-Tomography (USCT) is based on circa 2000 ultrasound-transducers, which are arranged in layers on a cylindrical tank filled with water. For one image, a huge amount of raw data of approximately 20 GB is recorded and stored to a hard disk. A special software reconstructs a volume out of the raw data.

In the current phase of developement it is necessary for the scientists to get a overview on the data as easy as possible and to conceive a complete view on the whole system. Until now they get only views on subareas of the USCT by the currently used software, have to analyse data manually and have only few possibilities to visualize the computed data. For this reason, a new software called „DiagTool“ is being developed within this diploma thesis. It allows the USCT working group the integrated exploration, analysis and visualization of the three main views on the USCT: the raw data, the reconstructed images and the coherence with the source of both - the appearance of the USCT.

The software is implemented in MATLAB and consists of three graphical user interfaces according to the three views on the USCT, which can be used independent as well as associated to each other. The user receives different possibilities for interaction as for example an interactive, dynamically generated three dimensional modell of the USCT. A programmable left mouse button speeds up and simiplifies the act of selecting sender or receiver elements. For the expandability of the software, a new interface concept is created. Despite of its numerous functions, the graphical user interface of the software is clearly arranged.

The new graphical representation of the available information and the connection between different views on the USCT provides completely new possibilities to gather the coherences easy and fast.

Inhaltsverzeichnis

Kapitel 1

Einleitung

1.1 Brustkrebsfrüherkennung

Brustkrebs ist in Deutschland, wie auch weltweit, die häufigste Krebserkrankung bei Frauen. Jede elfte Frau erkrankt im Laufe ihres Lebens daran. Bei einem Mammakarzinom liegt die Sterblichkeit bei 18% und nimmt somit den ersten Platz unter den Krebserkrankungen ein [GHJL05].

Die Überlebenschancen können wesentlich gesteigert werden, wenn eine Erkrankung bereits in einem frühen Stadium erkannt wird und somit die Wahrscheinlichkeit einer Metastasierung geringer ist.

Zur Brustkrebsfrüherkennung werden verschiedene bildgebende Verfahren eingesetzt:

Mammographie Bei der Mammographie handelt es sich um eine Untersuchung mit Röntgenstrahlung, bei der die Brust zwischen zwei Plexiglasplatten komprimiert wird, um die Strahlendosis möglichst gering zu halten [Wik06b]. Insbesondere bei jungen Frauen mit einem hohen Anteil an Drüsengewebe liefert diese Methode schlechte Ergebnisse. Zusätzlich zur Strahlenbelastung hat dieses Verfahren den Nachteil, durch die Kompression und die zweidimensionale Abbildung eine Lokalisierung eines möglichen Tumors zu erschweren.

Magnet-Resonanz-Tomographie (MRT) Die Magnet-Resonanz-Tomographie - meist auch synonym mit Kernspintomographie oder kurz „Kernspin“ bezeichnet - nutzt magnetische Felder, um Schnittbilder des menschlichen Körpers zu erstellen, welche zu einem dreidimensionalen Bild zusammengefasst werden können. Der Vorteil dieses Verfahrens ist die gute Bildqualität, womit Tumore schon ab einem sehr geringen Durchmesser von etwa 3 mm erkannt werden können. Allerdings sind MRT-Untersuchungen sehr teuer und haben eine niedrige Spezitivität weshalb sie nicht als Standardinstrument in der Brustkrebsfrüherkennung zu beurteilen sind [Wik06a].

Sonographie Mit Sonographie wird die Anwendung von Ultraschall als bildgebendes Verfahren bezeichnet. Untersuchungen zur Brustkrebsvorsorge werden mit einem Ultraschallkopf durchgeführt, der in das Gewebe einstrahlt. Meist erfolgt eine Ultraschalluntersuchung erst nach Hinweisen auf verändertes Gewebe durch andere Verfahren. Der Vorteil der Sonographie ist, dass für die Patientin keine Strahlenbelastung entsteht [Wik06c].

1.2 3D Ultraschall-Computertomographie

Am Institut für Prozessdatenverarbeitung und Elektronik (IPE) des Forschungszentrums Karlsruhe (FZK) wird mit der 3D Ultraschall-Computertomographie ein neues bildgebendes Verfahren zur Brustkrebsfrüherkennung entwickelt, das echt dreidimensionale Volumenbilder in wesentlich besserer Bildqualität als konventionelle Ultraschall-Verfahren ermöglichen soll. Durch die Verwendung von Ultraschall entsteht für die Patientin keine Belastung durch ionisierende Strahlung. Durch die dreidimensionale Abbildung wird eine Erkennung und Lokalisierung eines Tumors verbessert.

Die 3D Ultraschall-Computertomographie basiert auf einer Vielzahl von Ultraschallwandlern, die ringförmig auf einem wassergefüllten Zylinder angeordnet sind (Abbildung 1.1). Um eine möglichst hohe Bildqualität mit Auflösungen von ca. 0,1 mm zu erreichen ist eine sehr große Anzahl von Informationen aus allen Richtungen erforderlich.

Zur Gewinnung dieser Informationen sendet jeweils ein Ultraschallwandler, während alle anderen empfangen. Aus der Kombination aller Sender und Emp-

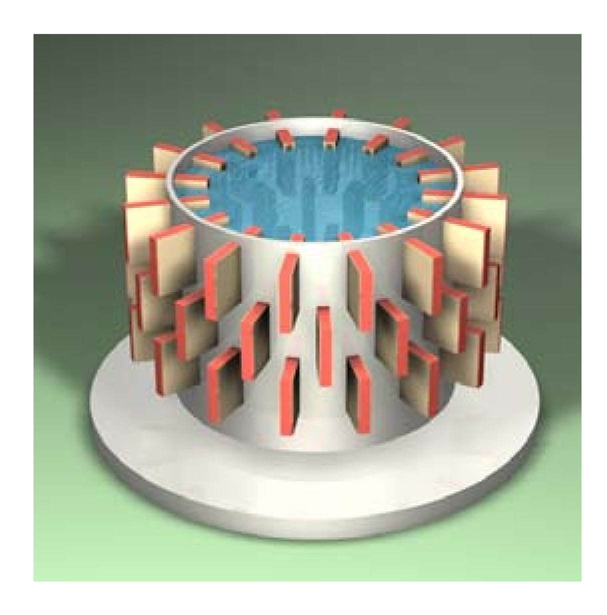


Abbildung 1.1: Modell des 3D Ultraschall-Computertomographen

fänger ergibt sich somit für eine Aufnahme eine eben so große Anzahl aufgenommener Datensätze mit entsprechendem Speichervolumen im Gigabyte-Bereich.

Anschließend wird aus den Rohdaten das Bild rekonstruiert. Hierfür wird das Prinzip der Reflexionstomographie verwendet, das aus Laufzeitinformationen der Ultraschallsignale die Objekte innerhalb des Ultraschall-Computertomographen (USCT) abbildet.

Eine ausführliche Einführung in die Ultraschall-Computertomographie wird in Abschnitt 4.1 gegeben.

1.3 Motivation

Bei einer Messung mit dem USCT werden ca. 3,5 Millionen Datensätze von ca. 2000 Ultraschallwandlern erfasst, die als Rohdaten auf der Festplatte des Datenakquisitionsrechners abgelegt werden. Anschließend wird aus den aufgenommenen Daten mit der Rekonstruktionssoftware, basierend auf der mathematischen Software MATLAB[1], das Bild rekonstruiert.

Sowohl bei der Rekonstruktion als auch bei der Datenaufnahme gibt es eine Vielzahl von Einflussmöglichkeiten und Parameter, die optimiert werden müssen, um die gewünschte Bildqualität zu erreichen. In der derzeitigen Entwicklungsphase analysieren die Wissenschaftler der Arbeitsgruppe USCT[2] die auftretenden Fehler und suchen mögliche Ursachen, die dann schließlich Hinweise auf Verbesserungsmöglichkeiten der unterschiedlichen Komponenten des Systems liefern.

Ein großer Nachteil dabei ist, dass viele einzeln vorhandene Programme und Skripte meist nur als Konsolenanwendungen entwickelt werden, da der Aufwand für die Gestaltung einer eigenen Benutzeroberfläche recht aufwendig ist. Oft erhalten diese eine Vielzahl unterschiedlicher Parameter, wodurch die Benutzung für andere Personen erschwert wird. Die sehr große Komplexität des Gesamtsystems trägt zum erschwerten Verständnis der Software bei. Eine zeitaufwendige Einarbeitung ist daher erforderlich.

Die Exploration der großen Datenmengen wird derzeit manuell ausgeführt, so dass Datensätze einzeln geöffnet werden müssen und die Anzeige der Daten erst nach Eingabe einer Befehlsfolge möglich ist. Die gezielte Auswahl von Datensätzen erfordert einen hohen zeitlichen Aufwand.

Die derzeitige Sichtweise erlaubt lediglich den Blick auf das Ergebnis einer Messung und das fertig rekonstruierte Bild. Der zwischen diesen beiden Enden der Verarbeitungskette befindliche Bereich kann im Nachhinein nur noch schwer und unter hohem Aufwand nachvollzogen werden.

Die Folgerung aus der beschriebenen Ist-Situation ist die Notwendigkeit einer Software, die den wissenschaftlichen Mitarbeitern eine grafische Benutzeroberflä-

[1] The MathWorks Inc.: MATLAB, `http://www.mathworks.com`

[2] Arbeitsgruppe Ultraschall Computertomographie am Institut für Prozessdatenverarbeitung und Elektronik im Forschungszentrum Karlsruhe. Webseite: `http://fuzzy.fzk.de/~USCT`

che (GUI - für Graphical User Interface) bietet, mit der es möglich ist auf den Rohdaten wie auch auf den rekonstruierten Bildern zu arbeiten und Analysen der Signale, Bilder usw. durchzuführen. Benötigt werden Möglichkeiten zur schnellen grafischen Darstellung von Informationen, um einen besseren Überblick über die großen Datenmengen, deren Herkunft und Wirkung zu verschaffen. Ein ebenso wichtiger Punkt ist die Verknüpfung der Rohdaten mit den Daten rekonstruierter Bilder, um die Ursachen von auftretenden Problemen zu ergründen. Für verschiedene sich wiederholende Arbeitsabläufe soll eine Automatisierung erfolgen.

Dabei ist es unbedingt erforderlich auf die Erweiterbarkeit und Anpassbarkeit der Software zu setzen, da sich durch neue Erkenntnisse ständig neue Anforderungen an die Analysemethoden ergeben. Für zukünftige Entwicklungen soll es einfach sein eine grafische Oberfläche zu benutzen, um den Umgang mit Analysefunktionen für alle Mitarbeiter zu erleichtern. Bisherige alleinstehende Programme sollen in einer gemeinsamen Benutzeroberfläche integriert werden.

Kapitel 2

Aufgabenstellung

Im Rahmen der vorliegenden Diplomarbeit soll eine erweiterbare Software erstellt werden, die es den wissenschaftlichen Mitarbeitern der USCT-Arbeitsgruppe ermöglicht, die großen Datenmengen einer USCT-Messung zu durchsuchen, zu visualisieren und zu analysieren.

Im Folgenden wird auf die Anforderungen an die Software näher eingegangen.

2.1 Funktionale Anforderungen

Die bei einer Messung mit dem USCT entstehenden Rohdaten werden als A-Scans (Amplituden-Scan) bezeichnet. Die zu erstellende Software soll das Laden und Plotten der selbigen anhand der Angabe des Sende- und Empfangselements sowie des Experimentverzeichnisses ermöglichen. Dabei muss die Struktur berücksichtigt werden, in der die Daten vorliegen (vergl. hierfür Kapitel 4.1.3). Über jedes Experiment und jeden A-Scan soll es möglich sein, Informationen, die während einer Messung gespeichert werden, anzuzeigen. Genannt seien hier nur beispielhaft der Name des Experimentes, die Parameter mit denen die Messung initialisiert wurde oder die Nummer des jeweils ausgewählten Senders und Empfängers.

Eine weitere Anforderung ist die Möglichkeit des Vergleichs von Daten unterschiedlicher Sender-Empfänger-Kombinationen durch die gleichzeitige Anzeige mehrerer A-Scans. Die weiterverarbeiteten Signaldaten sollen als neuer Rohdatensatz abgespeichert werden können. Ebenso ermöglicht werden muss das Exportieren des Plots in eine Bild-Datei im JPEG- oder PNG-Format.

Erstellt werden soll eine Schnittstelle für die Einbindung von auf Rohdaten arbeitenden Transformationen. Über diese wird es dem Benutzer ermöglicht, der Software selbstständig Funktionen hinzuzufügen. Eine Ausführung der Transformationen wird auch auf Teilbereichen eines Datensatzes, d.h. markierten Ausschnitten ermöglicht.

Ein wichtiger Punkt in der Analyse der Daten ist die Verbindung zwischen

den Rohdaten und den rekonstruierten Bildern. Die Software soll es daher ermöglichen eine Rekonstruktionsellipse (zur Erläuterung der USCT-Grundlagen sei an dieser Stelle auf Abschnitt 4.1 verwiesen) anhand eines markierten Bereiches im A-Scan zu berechnen und zu visualisieren. Den entgegengesetzten Weg geht die Auswahl eines Bildbereiches, von dem auf einen Bereich im A-Scan zurückgerechnet werden soll. Letztere Funktion soll ebenfalls auf mehreren ausgewählten A-Scans ausführbar sein, wodurch z.B. bestimmt werden kann, welche A-Scans am meisten zu einem Bildbereich beigetragen haben.

Wie bereits zu erkennen ist, beschäftigen sich einige weitere Funktionen mit den Daten rekonstruierter Bilder. Zunächst soll es ermöglicht werden diese zu laden und anzuzeigen. Die Funktion soll den Vergleich zweier Bilder ermöglichen, ebenso wie die Ausführung beliebiger Bildvergleichs- bzw. Bildzusammenführungsberechnungen. Letztere Funktion soll ähnlich wie die A-Scan-Transformationen über ein Pluginkonzept für das Einhängen selbst programmierter Funktionen erweiterbar sein.

Für die rekonstruierten Bilder ist es erwünscht Bildverarbeitungsfunktionen anzubieten. Darunter fällt z.B. die Anpassung der Helligkeit eines Bildes oder die Veränderung der MATLAB-Colormap. Des Weiteren soll eine logarithmierte Darstellung der Daten auswählbar sein.

Der Export der Bilder soll sowohl im MATLAB-Datenformat wie auch im JPEG-Bildformat gewährleistet werden.

Informationen über ein rekonstruiertes Bild bezüglich der Größe, dem Aufnahmemodus usw. sollen in der zu erstellenden Benutzeroberfläche dargestellt werden.

Für Volumenbilder soll eine Möglichkeit der Visualisierung entwickelt werden, mit der der Benutzer ein Volumen in generierten Schnittbildern durchlaufen kann. Die Schnittebene ist dabei aus den drei Hauptachsenkombinationen XY, XZ und YZ auswählbar zu gestalten.

Zur Gewährleistung der Erweiterungsfähigkeit der Software im Bereich der rekonstruierten Bilder soll eine weitere Schnittstelle geschaffen werden, über die das Einhängen beliebiger Funktionen, die auf Ausschnitten des Bildes operieren, möglich ist.

Ein weiterer Teil der Anforderungen beschäftigt sich mit der Abbildung des Ultraschall-Computertomographen in einem interaktiven Modell, in dem Sender und Empfänger mit der Maus markierbar sein sollen. Auswahlbereiche aus der Volumenvisualisierung (so genannte ROIs - Regions of Interest) sollen im USCT-Modell an der entsprechenden Stelle dargestellt werden, um die Herkunft des dargestellten Bildes räumlich zuzuordnen.

Anhand der markierten Sende- und Empfangselemente soll die Generierung von Parameterdateien für andere mit dem USCT verbundene Programme ermöglicht werden. Textdateien mit der linearen Durchnummerierung der gewünschten Sender und Empfänger werden von der Datenakquisitionssoftware benötigt,

um eine Messung z.B. lediglich mit einer Teilmenge der vorhandenen Ultraschallwandler zu starten, wodurch sich die Aufnahmezeit verkürzt. Eine Datei im MATLAB-Datenformat gibt der Rekonstruktionssoftware an, welche Sender-Empfänger-Kombinationen bei der Messung vorhanden waren. Die Generierung einer solchen Datei soll eine Rekonstruktion mit speziellen bzw. eingeschränkten Sender-Empfänger-Kombinationen erlauben.

Als allgemeine Erweiterungsmöglichkeit soll eine Schnittstelle zum Einhängen beliebiger Funktionen geschaffen werden. Die Programmierer solcher Funktionen sind dabei frei in der Wahl der Implementierungs- und Visualisierungsart. Sie können sowohl gänzlich unabhängig von der Software arbeiten, aber auch die Visualisierungsmöglichkeiten der Software, wie z.B. das USCT-Modell zur Anzeige von Ergebnissen verwenden.

Allgemein ist zu sagen, dass die Funktionen jeweils verzahnt gesehen werden sollen, d.h. Änderungen in Teilbereichen der grafischen Oberfläche wirken sich auch auf andere Teilbereiche aus. Näher genannt sei hier nur als Beispiel die Verbindung des USCT-Modells mit der Auswahlmöglichkeit für A-Scans.

2.2 Allgemeine Anforderungen

Die genannten funktionalen Anforderungen sollen in einer grafischen Benutzeroberfläche zur Verfügung gestellt werden, die einen Überblick über alle Teilbereiche des Ultraschall-Computertomographen liefert: sie stellt Verbindungen zwischen den bisher recht isoliert betrachteten Sichtweisen auf rekonstruierte Bilder, Rohdaten und physikalischen Aufbau her und integriert dabei vorhandene Funktionen in einer gemeinsamen grafischen Umgebung.

Dem Benutzer soll eine möglichst intuitive Bedienung geboten werden, die es ihm ermöglicht mit maximal vier Mausklicks die wichtigsten Funktionen zu erreichen. Aufgrund der Internationalität der USCT-Abteilung ist die Benutzeroberfläche mit englischen Texten zu versehen.

Ein wichtiger Punkt ist die Erweiterbarkeit der Software. Wie in den funktionalen Anforderungen beschrieben, sollen eigene Funktionen in die Software eingebunden werden können. Dies erfolgt durch die Erstellung von Schnittstellenspezifikationen, deren Konventionen der Programmierer einhalten muss.

Da es sich bei der Entwicklungsumgebung MATLAB um eine Skriptsprache handelt, die beispielsweise ein Package-Konzept, wie es aus Hochsprachen bekannt ist, nicht besitzt, ist es notwendig eine geeignete Strukturierung der MATLAB-Funktionen in Ordnern und Unterordnern aufzubauen.

Die Software ist lauffähig auf allen Systemen zu gestalten, die Matlab in der Version 6 oder höher installiert haben.

Kapitel 3

Verfahren und Methoden

Zum Erreichen der Aufgabenstellung steht die folgende Hard- und Software zur Verfügung:

3.1 Hardware

Als Entwicklungsumgebung dient ein Rechner der Marke Fujitsu Siemens mit einem 3,2 GHz Pentium IV Prozessor, 1024 MB Arbeitsspeicher und 80 GB Festplatte. Der Prozessor unterstützt Hyper-Threading, das eine zweite virtuelle Prozessoreinheit zur Verfügung stellt, um den Prozessor effizienter zu nutzen. Zur Speicherung der Software wird über ein Fast Ethernet Netzwerk zusätzlich auf den Fileserver des Instituts für Prozessdatenverarbeitung und Elektronik zurückgegriffen. Als Bildschirm dient ein 18,1" LCD-Flachbildschirm der Marke EIZO mit einer Auflösung von 1280×1024 Pixel.

Die Datensätze, auf denen gearbeitet wird, stammen aus dem am IPE entwickelten Prototypen des 3D Ultraschall-Computertomographen und sind auf oben benanntem Fileserver abgelegt. Ebenso finden sich dort die Daten der rekonstruierten Bilder, die auf verschiedenen Rechnern der Arbeitsgruppe berechnet wurden.

3.2 Betriebssystem

Als Betriebssystem wird auf dem Entwicklungsrechner Windows XP Professional mit Service Pack 2 verwendet, auf dem regelmäßig Updates installiert werden. Der 3 GB Arbeitsspeicher-Switch ist in der Systemdatei boot.ini aktiviert, um Programmen einen virtuellen Arbeitsspeicher von 3 GB anstatt der üblichen 2 GB zu ermöglichen, was besonders für speicherintensive MATLAB-Anwendungen relevant sein kann. Dieses Betriebssystem wird ebenfalls auf den anderen Rechnern der Arbeitsgruppe verwendet. Des Weiteren wird teilweise Suse Linux 10.0 in einer Parallelinstallation angeboten.

Zum Testen der Software stehen zuletzt genannte Rechner ebenfalls zur Verfügung, welche unterschiedliche Hardware-Konfigurationen bieten.

3.3 Software

Primär verwendet wird die Software MATLAB. Diese ist in der Version 6.1 und 7.1 auf dem Entwicklungsrechner installiert, um bei der Entwicklung die Kompatibilität der Software zwischen den verschiedenen Versionen zu gewährleisten. Um Berechnungen - unter anderem für die Bildrekonstruktionssoftware - zu beschleunigen wird der Matlab Performance Pack v.1.11 benutzt. Dieser passt Matlab auf den Prozessortyp des Rechners an und beschleunigt somit die Ausführung von Algorithmen. Die Testrechner sind mit den gleichen Matlab-Versionen ausgestattet.

Verwendete MATLAB-Bibliotheken stammen von der Webseite MATLAB File Exchange[1] und werden im Folgenden separat referenziert.

Zur Erstellung von Grafiken für die Benutzeroberfläche wird das Bildbearbeitungsprogramm „Jasc Paint Shop“ in der Version 8.05 verwendet.

Zur Erzeugung von Testdatensätzen für die Rohdaten wird die Datenakquisitionssoftware „Andromeda“ verwendet, die am IPE entwickelt wurde. Die aktuelle Version trägt die Bezeichnung 2.1. Zur Rekonstruktion von Bildern dient die aktuelle Version 4.3.1 der Bildrekonstruktionssoftware. Sie wurde ebenfalls am IPE entwickelt und basiert auf MATLAB, wodurch einzelne Funktionen entsprechend angepasst in das Datenanalyse-Tool einfließen können.

Eine Versionierung der Software findet über das Subversion-System (SVN) der USCT-Abteilung statt, für das als Windows-Client die Software Tortoise SVN in der Version 1.4.0 verwendet wird.

Die vorliegende Diplomarbeit wird mit der TEX-Implementierung MIKTEX 2.4 erstellt. Als Editor dient die kostenlose Software TeXnicCenter 1 Beta 6.31 ('Firenze'). Für Grafikerstellung und Verwaltung von zusätzlichen Textdokumenten dient Microsoft Office XP mit den aktuellsten Updates.

Des Weiteren wird zur Erstellung von Grafiken Macromedia Fireworks MX 2004 (Version 7.0.2.295) sowie Macromedia FreeHand MX (Version 11.0.2) verwendet.

[1] MATLAB File Exchange: `http://www.mathworks.com/matlabcentral/fileexchange/loadCategory.do`

Kapitel 4

Grundlagen

4.1 3D Ultraschall-Computertomographie

Bei der 3D Ultraschall-Computertomographie handelt es sich um ein neuartiges bildgebendes Verfahren, das am Institut für Prozessdatenverarbeitung und Elektronik im Forschungszentrum Karlsruhe entwickelt wird. Nachdem zunächst die Machbarkeit mit einem 2D-Demonstrator nachgewiesen wurde [Wür00], wird zurzeit an der Entwicklung des 3D-Prototyps geforscht. Im Folgenden soll das Prinzip der Ultraschall-Computertomographie, der Aufbau des Prototyps sowie die Verfahren zur Aufnahme und Rekonstruktion von Bildern näher erläutert werden.

4.1.1 Prinzip der Ultraschall-Computertomographie

Das Prinzip der Ultraschall-Computertomographie (Abbildung 4.1) beruht auf dem Einstrahlen von Ultraschallwellen in einen wassergefüllten Zylinder. Nachdem ein Sendeelement einen Ultraschallimpuls ausgesendet hat (gelbe Linien) zeichnen alle Empfangselemente die durch den Zylinder gelaufene und mit dem Objekt interagierte Ultraschallwelle auf.

Befinden sich Objekte innerhalb des Messzylinders (in der Abbildung der rote Punkt im Zentrum des Zylinders), werden die Ultraschallwellen je nach Beschaffenheit und Material des Objektes an ihm reflektiert, absorbiert oder gebrochen (schwarze Linien). Die Auswertung der aufgenommenen Signale ermöglicht die Rekonstruktion der Objekte. Aufgrund der Informationen, die aus allen Richtungen gewonnen werden, können detaillierte dreidimensionale Bilder erzeugt werden.

Bei der ärztlichen Untersuchung umschließt der USCT die Brust der liegenden Patientin vollständig. Sie hängt damit frei und ohne Deformation nach unten, wodurch auffällige Strukturen auf den erzeugten Bildern in der tatsächlichen Brust besser lokalisiert werden können. Die Ultraschallwandler sind um die gesamte

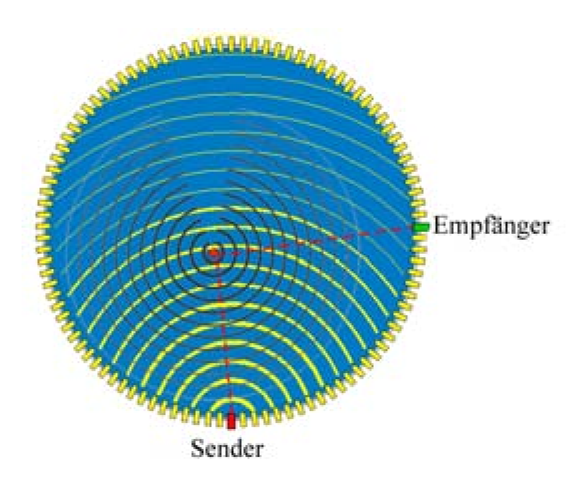


Abbildung 4.1: Prinzip der Ultraschall-Computertomographie

Brust herum angeordnet, wodurch entsprechend viele Informationen von allen Seiten gesammelt werden können.

4.1.2 Physikalischer und technischer Aufbau

Der Ultraschall-Computertomograph (Abbildung 4.2) besteht aus einem Metallzylinder, der mit dem verwendeten Koppelmedium Wasser gefüllt ist. Der Durchmesser beträgt ca. 20 cm, die Höhe etwa 23 cm.

Ringförmig sind an der Innenwand des Zylinders in drei Ebenen sogenannte Transducer Array Systeme (TAS) - mehrere in einem Bauelement untergebrachte Ultraschallwandler - angebracht. Jede Ebene (auch als TAS-Ring bezeichnet) besitzt 16 TAS, wodurch sich ein Abstand zwischen zwei TAS von $\frac{360°}{16} = 22,5°$ ergibt. Insgesamt besitzt der USCT $16 \cdot 3 = 48$ TAS. Der mittlere TAS-Ring ist aus bautechnischen Gründen um 11,25°versetzt.



Abbildung 4.2: Der Ultraschall-Computertomograph. Der mittlere Ring ist mit Transducer Array Systemen bestückt.

Jedes TAS (Abbildung 4.3) besitzt 8 übereinander angeordnete Sendeelemente mit einem vertikalen Abstand von 6 mm. Umgeben ist ein Sender von jeweils 4 Empfangselementen, die sich zu einer Gesamtzahl von 32 pro TAS addieren. Der horizontale wie auch der vertikale Abstand zwischen einem Sende- und einem Empfangselement beträgt 1,5 mm. Multipliziert mit der Anzahl der TAS ergibt sich eine Gesamtzahl von 384 Sendern und 1536 Empfängern.

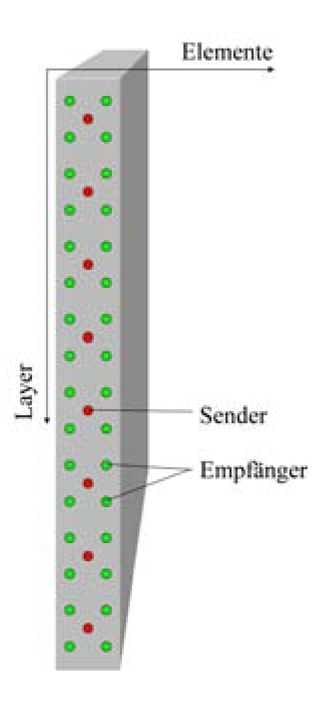


Abbildung 4.3: Transducer Array System des Ultraschall-Computertomographen

Um eine gleichmäßige Verteilung der Sender und Empfänger auf der Zylinderoberfläche zu erreichen, besitzt der USCT einen Schrittmotor, der den auf einem Kugellager drehbar gelagerten Zylinder auf 6 verschiedene Positionen drehen kann. Der Abstand zwischen zwei Motorpositionen beträgt 3,75°. Virtuell kann somit die Gesamtzahl auf 2304 Sender und 9216 Empfänger gesteigert werden [Hop05].

Die logische Durchnummerierung zur Identifikation und Ansteuerung der Elemente unterteilt den USCT in Schichten von Sendern (sogenannten Senderlayern), Sendernummern sowie Empfängerlayer und Empfängernummern. Beginnend von oben gibt es somit 3 TAS-Ringe · 8 Sendeelemente pro TAS = 24 Sendelayer. Anhand der rechteckigen Anordnung der Empfänger um einen Sender herum ergeben sich pro TAS 16 und damit als Gesamtzahl 48 Empfänger-Layer. Pro Senderlayer gibt es 96 Sendernummern, die sich aus der Anzahl der Motorposition (6) multipliziert mit der Anzahl der TAS (16) ergeben. Bei den Empfangsnummern folgt

analog eine Anzahl von 192.

Die Nummerierung beginnt in diesem System bei 0. Aufgrund der MATLAB-Eigenschaft Arrays und Matrizen beginnend mit der 1 zu indizieren, ist es daher des öfteren erforderlich ein Mapping zwischen beiden Varianten durch Addition bzw. Subtraktion von 1 durchzuführen.

Des Weiteren gibt es eine lineare Durchnummerierung für Sender und Empfänger. Beginnend beim Element 0 aus Layer 0 steigt die Nummerierung gegen den Uhrzeigersinn an und wird beim Sprung zum nächsten Layer nicht zurückgesetzt. Daraus ergeben sich real addressierbare Sendeelemente (ohne Berücksichtigung der Motorpositionen) von 0 bis 383 und Empfängerelemente von 0 bis 1535. Unter Hinzunahme von Motorpositionen ändert sich die Nummerierung auf Senderelemente von 0 bis 2303 und Empfängerelemente von 0 bis 9215.

Die räumlichen Positionen der Sender und Empfänger werden in einem kartesischen Koordinatensystem beschrieben, dessen Ursprung im Zentrum des USCT an der Oberkante des obersten TAS-Ringes liegt (Abbildung 4.4). Die Koordinaten werden in den so genannten Geometrie-Dateien jeweils für Sender und Empfänger gespeichert.

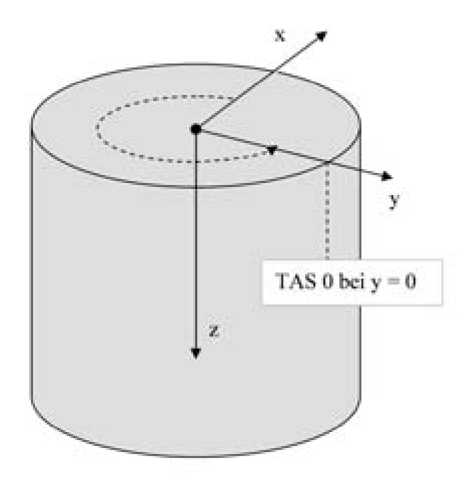


Abbildung 4.4: Lage des kartesischen Koordinatensystems

Zur Dämpfung der Ultraschallwellen an den Wänden des USCT-Zylinders wird eine Dämpfungsschicht aus Polyurethan verwendet. Dies erfordert eine leichte Verschiebung der TAS ins Innere des Zylinders, wodurch sich der effektive Durchmesser auf ca. 18 cm verringert.

4.1.3 Datenaufnahme

Die Datenaufnahme erfolgt über die Datenakquisitionssoftware „Andromeda“, eine in Java implementierte Benutzerschnittstelle ähnlich einer Shell, welche über

Dynamic Linked Libraries (DLL) Zugriff auf die Hardware hat.

Hardwaresicht

Eine Messung mit dem Ultraschall-Computertomographen wird von Andromeda initiiert. Im Datenakquisitionsrechner ist eine microEnable-Karte[1] eingebaut, die Steuerbefehle von Andromeda an die zweite Hardwarestufe, das USCT-Crate weitergibt.

Dort kontrolliert das SLT (Second Level Triggerboard) sämtliche Abläufe innerhalb des Crates und des PBuses (Protokoll-Bus), über den Daten und Steuerbefehle übertragen werden. Das Control Board (CTR) adressiert über einen seriellen Bus die TAS und erzeugt das analoge Sendesignal.

Dieses an alle Sender gesendete Signal wird in den Distribution Boards einem Multiplexing unterzogen, so dass der Anregungspuls für die Ultraschallerzeugung nur an genau ein TAS übertragen wird. Das nun ausgewählte TAS sendet mit einem der acht Sendeelemente das Ultraschallsignal in den Messzylinder aus.

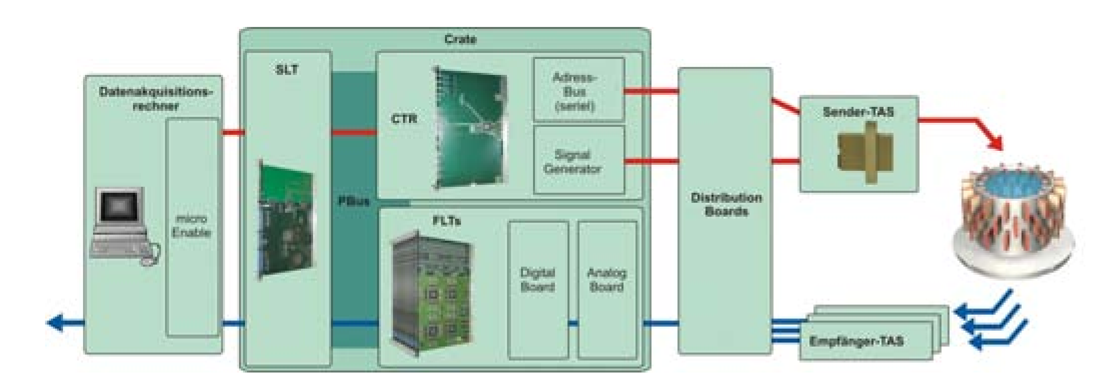


Abbildung 4.5: Überblick über die Hardware des USCT für die Datenaufnahme

Gleichzeitig nehmen alle Empfänger das Signal auf, wobei die Aufnahmedauer 3 μs beträgt. Die Signale werden zurück an die Distribution Boards gesendet. Aufgrund der hohen Empfänger-Anzahl wird bereits in den TAS ein Multiplexing durchgeführt, so dass jeweils 4 Signale parallel weitergeleitet werden. Bei insgesamt 32 Empfängern pro TAS müssen zum Auslesen sämtlicher Daten also 8 Messdurchgänge mit dem gleichen Sender durchgeführt werden.

Die Distribution Boards sammeln die Daten aller TAS und senden diese über insgesamt 192 parallele (48 pro Board) Analogkanäle an die First Level Triggerboards (FLT). Dort werden die Signale zunächst analog gefiltert und verstärkt bevor die Analog-Digitalwandlung mit 12 Bit Quantisierung stattfindet. Dabei wird eine Abtastfrequenz von 10 MHz verwendet. Aus der Aufnahmedauer von

[1]microEnable-Karte: Schnittstellenkarte zur Verbindung von Elektronik und Rechner. Der microEnable Coprozessor ist komplett programmierbar

300 μs lässt sich die Gesamtzahl der Abtastpunkte (Samples) eines einzelnen A-Scans nach Formel (4.1) errechnen.

$$0,0003\,s \cdot 10000000 \cdot 1/s = 3000\,Samples \tag{4.1}$$

Die FLTs speichern die Daten zwischen, bis sie vom SLT über den PBus ausgelesen werden und von dort an die microEnable-Karte des Datenakquisitionsrechners gesendet werden. Der Rechner erfasst die digitalen Messdaten und speichert sie als A-Scans ab.

Ein Überblick über die Durchführung einer Messung aus Hardwaresicht verschafft Abbildung 4.5.

Softwaresicht

Wie bereits erwähnt wird eine Messung mit dem USCT von der Steuersoftware Andromeda initiiert. Bevor eine Messung gestartet wird, müssen einige Parameter angegeben werden.

Zunächst ist es erforderlich einen Ordner auf der Festplatte zu erstellen, in dem alle zum Experiment gehörenden Dateien abgelegt werden.

Andromeda muss eine Liste der Sender und Empfänger vorliegen, die bei der Messung angesteuert werden sollen. Diese Listen werden in zwei Textdateien „AvailableEmitters.txt“ und „AvailableReceivers.txt“ abgelegt. Die Datei startet mit einem auskommentierten Header, wobei das Zeichen '#' eine Zeile der Textdatei als Kommentar deklariert. In der ersten unkommentierten Zeile stehen darauf folgend die realen linearen Sendernummern durch ein Leerzeichen getrennt (siehe Anhang B.1). Der Aufbau der Datei für die Empfänger verläuft analog (Anhang B.2).

In der Datei „codedExcitation.txt“ wird der Anregungspuls für die Ultraschallwandler beschrieben. Auch hier dient das Raute-Zeichen '#' als Einleitung für eine Kommentarzeile. Darauffolgend stehen 510 16-Bit-Zahlenwerte in Dezimalnotation (Anhang B.3).

Die Datei „Bitfield.mat“ enthält ein Array mit 2.654.208 Elementen à 8 Bit, die die insgesamt 21.233.664 theoretisch möglichen Sender-Empfänger-Kombinationen repräsentieren. Soll eine Messung eine Kombination verwenden, ist das entsprechende Bit auf 1, andernfalls auf 0 gesetzt. Die Position eines Bits n kann wie folgt einer Sender-Empfänger-Kombination zugeordnet werden:

$$SL = 1 + \frac{n}{anzSN \cdot anzRL \cdot anzRN} \quad \text{mit Rest} \quad restSL \tag{4.2}$$

$$SN = 1 + \frac{restSL}{anzRL \cdot anzRN} \quad \text{mit Rest} \quad restSN \tag{4.3}$$

$$RL = 1 + \frac{restSN}{anzRN} \quad \text{mit Rest} \quad restRL \tag{4.4}$$

$$RN = restRL \tag{4.5}$$

mit SL = Senderebene, SN = Sendeelement, RL = Receiverebene, RN = Receiverelement, sowie der Gesamtanzahl der verfügbaren Sendeelemente (anzSN, im USCT derzeit 96), Receiverebenen (anzRL, derzeit 48) und Receiverelemente (anzRN, derzeit 192) [MR05]. Bei den Divisionen handelt es sich um ganzzahlige Divisionen. Die Bitposition n beginnt bei 1.

Andromeda startet eine Messung durch Eingabe des Befehls „scan“. Es werden nun die beschriebenen Dateien eingelesen und die Messung initiiert. Nach dem aufgezeigten hardwareseitigen Messverlauf wird ein Datenarray mit 3000 Elementen (pro Sample ein Element) zurückgeliefert, das als A-Scan bezeichnet wird. Arrayelement n enthält die gemessene Amplitude zum Aufnahmezeitpunkt $t = n \cdot 0,0000001\ s$.

Die empfangenen Daten werden auf der Festplatte in dem vor Beginn der Messung angegebenen Ordner abgelegt. Bei der Verarbeitung wird von Andromeda bereits das Matlab-Datenformat erzeugt. Für die Speicherung der Rohdaten stehen drei unterschiedliche Datenstrukturen zur Verfügung.

Single Modus Im Single-Modus wird jeder A-Scan in einer eigenen Matlab-Datei gespeichert. Als Variablenname innerhalb der Datei wird „AScan“ verwendet. Die Dateien werden in den Unterordnern für Senderebenen, Sendeelemente und Receiverebenen abgelegt (Abbildung 4.6). Der Dateiname setzt sich analog zu den Ordnernamen aus „receiverNumber_“ und der Nummer des Receiverelements in vierstelliger Notation mit führenden Nullen zusammen.

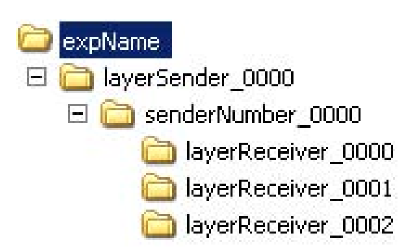


Abbildung 4.6: Ordnerstruktur des Single Datenformates

Multiple Modus Im Multiple-Modus werden alle zu einem Sendeelement gehörenden A-Scans gesammelt und gemeinsam in eine Matlab-Datei mit dem Namen „data.mat“ geschrieben. Für jeden A-Scan wird eine eigene Variable

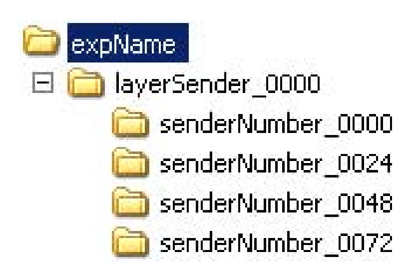


Abbildung 4.7: Ordnerstruktur des Multiple Datenformates

angelegt, die sich aus der Nummer der Receiverebene und des Receiverelements in der Form 'rlXrnX' zusammensetzt. Für die Senderebenen und Senderelemente wird die Ordnerstruktur analog zum Single Modus angelegt (Abbildung 4.7).

Lookup Modus Der Lookup-Modus sammelt wie auch der Multiple-Modus alle A-Scans zu einem Sendeelement. Die A-Scans werden jedoch in einer Variablen mit dem Namen „AScans" gespeichert, die als zweidimensionale Matrix aufgebaut ist. Die n Spalten entsprechen den jeweiligen Receiverelementen, die 3000 Zeilen den Samplepunkten des A-Scans. Die Speicherung der Nummern der Receiver erfolgt über eine ebenfalls erzeugte Matrix mit dem Namen „Map", die in zwei Zeilen für jeden A-Scan die Nummer der Receiverebene und des Receiverelements speichert. Die Zuordnung zwischen Daten und Receiver-Information erfolgt somit über den Spaltenindex. Die beiden Variablen werden in einer Datei „data.mat" gespeichert. Die Ordnerstruktur entspricht der des Multiple Modus.

Während der Messung werden neben den eigentlichen Daten auch zusätzliche Informationen abgespeichert. Die Datei „captains.log" dokumentiert den Verlauf einer Messung, beinhaltet dabei Fehlermeldungen und Warnhinweise. Senderspezifische Informationen werden in der Datei „emitterInfo.mat" im jeweiligen Unterordner des Sendeelements gesichert. Dies sind die durchschnittliche Temperatur sowie das Datum und die Zeit, zu der der Sendevorgang gestartet wurde.

Im obersten Verzeichnis der Messung werden allgemeine Informationen in der Datei „info.mat" bereitgestellt. Diese umfasst:

- den Experiment-Namen
- eine Beschreibung des Experimentes
- den Startpunkt der Aufzeichnung des Experimentes
- den Endpunkt der Aufzeichnung des Experimentes
- den Namen der verantwortlichen Person
- die verwendete Signalverstärkung

- die verwendete Abtastrate
- den Sample-Offset, der angibt wieviele Samples am Anfang eines A-Scans verworfen werden.
- die Anzahl der Samples pro A-Scan
- den elektronischen Offset
- die verwendete Frequenz des Tiefpassfilters
- die Anzahl der Samples einer vollen Wällenlänge
- die Versionsnummer der verwendeten Hardware
- die Bezeichnung der verwendeten Hardware
- die Versionsnummer der verwendeten Software
- den Namen des Kompressionsalgorithmus, der vor dem Speichern der A-Scans angewendet wurde
- die verwendete Eingangspulsform (Coded Excitation)
- der Modus für die Sender-Empfänger-Kombinationen (3D, 2D)
- den Namen der verwendeten Leermessung
- den Namen der Bitfield-Datei.

4.1.4 Bildrekonstruktion

Bei der Ultraschall-Computertomographie können verschiedene Informationen ausgewertet und daraus Bilder rekonstruiert werden:

- Reflexionsbilder, bei denen die Laufzeiten und Amplituden der Reflexionen ausgewertet werden.
- Absorptionsbilder, bei denen die Abschwächung der Signale beim Durchlaufen des USCT als Information genutzt werden.
- Schallgeschwindigkeitsbilder bei denen der Zeitpunkt des Eintreffens des ersten Pulses berücksichtigt wird.

Aktuell wird beim Versuchsaufbau am IPE die Reflexionstomographie zur Erzeugung von Bildern verwendet. Das Prinzip beruht auf der Zurückrechnung der Laufzeiten der Signale auf zurückgelegte Strecken.

Die Signallaufzeit t wird anhand des Sampleindexes und der Information über die Samplerate berechnet (Formel (4.6)).

$$t = \frac{1}{Samplerate} \cdot Sampleindex \tag{4.6}$$

In Abhängigkeit von der Temperatur breitet sich Schall im Medium Wasser mit unterschiedlicher Geschwindigkeit aus. Zur Bestimmung der Schallgeschwindigkeit v wird die Methode nach Marczak verwendet [Nat04].

Nun lässt sich sehr einfach die zurückgelegte Strecke s des Signals berechnen:

$$s = v \cdot t \tag{4.7}$$

Unter Annahme eines einzelnen Punktstreuers kann nun das an ihm reflektierte und vom Empfänger erfasste Signal auf eine zurückgelegte Wegstrecke umgerechnet werden.

Betrachtet man alle möglichen Wege der Länge s die der Schall von Sender $S1$ zu Empfänger $E1$ zurücklegen konnte, so ergibt sich eine Ellipse mit den Brennpunkten $S1$ und $E1$ (Abbildung 4.8), deren Linie die Menge aller möglichen Positionen des reflektierenden Objektes darstellt.

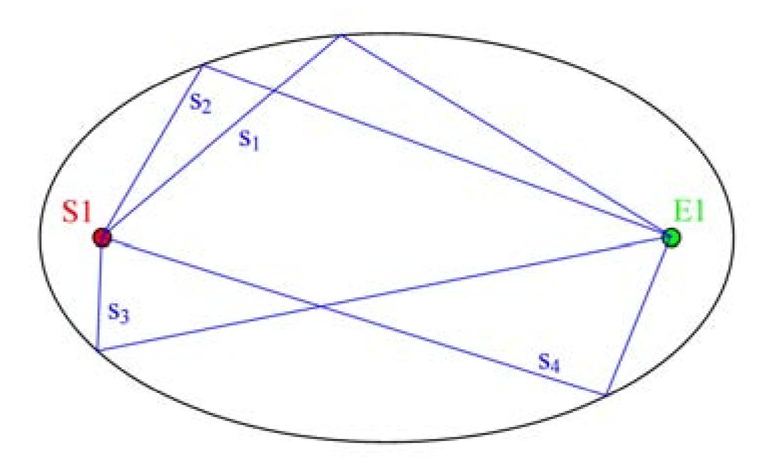


Abbildung 4.8: Sender S1 (rot), Empfänger E1 (grün) und Wege s_1, s_2, s_3, s_4 gleicher Länge (blau)

Für jeden Empfänger lässt sich somit für eine aufgezeichnete Reflexion an einem Samplepunkt des A-Scans eine eigene Ellipse berechnen. Der Punktstreuer befindet sich dann in dem Punkt, in dem sich die Ellipsen aller Empfänger schneiden (Abbildung 4.9).

Bei den erfassten A-Scans wird für die Rekonstruktion eines Bildes jeder einzelne Samplepunkt als Punktstreuer interpretiert und damit 3000 Ellipsen errechnet. Erweitert man nun das Prinzip auf alle vorhandenen Sender-Empfänger-Kombinationen und berücksichtigt die gemessene Amplitude (d.h. die Stärke des empfangenen Signals) an den jeweiligen Samplepunkten, so werden sich die für alle Samplepunkte erzeugten Ellipsen an Stellen, an denen sich stark streuende Objekte im USCT befanden, zu starken Intensitäten aufsummieren. Dieses Verfahren wird auch als Rückprojektion bezeichnet. Im dreidimensionalen Raum spricht man anstatt der Ellipsen von Ellipsoiden.

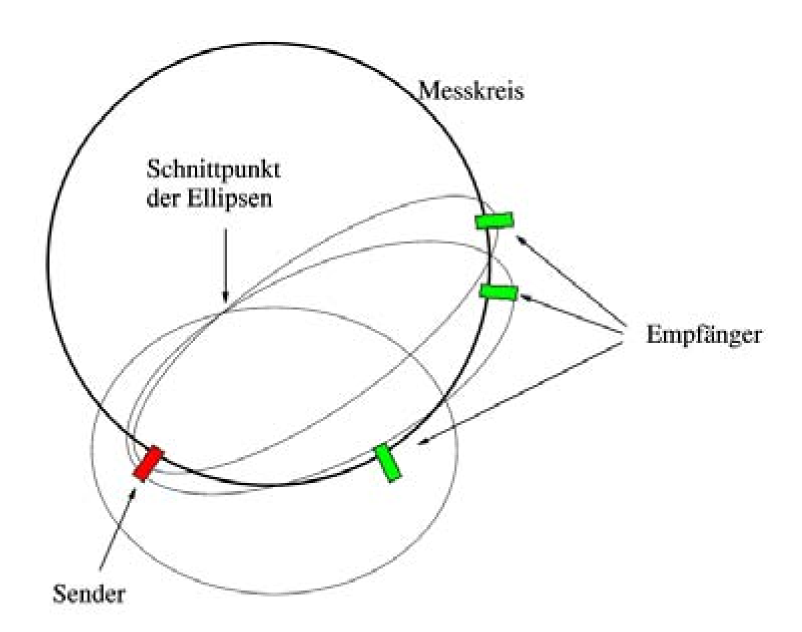


Abbildung 4.9: Rekonstruktionsprinzip beim USCT. Die Ellipsen schneiden sich im Streuer.

Die Rekonstruktion der Bilder aus den Rohdaten nach dieser Methode wird durch eine in MATLAB implementierte Software umgesetzt. Der Kern der Rekonstruktionssoftware - das Rückprojezieren von Ellipsoiden in ein Volumen - ist in C und x86-Assembler programmiert und wird über eine DLL angesprochen.

Zum Einstellen zahlreicher Parameter dient der Rekonstruktionssoftware die Datei „global3D.m“. In ihr werden unter anderem festgelegt:

- Von welchen Sendern und Empfängern die Signale für die Rekonstruktion verwendet werden sollen. Somit ist es auch möglich mittels einer Teilmenge aller möglichen Sender-Empfänger-Kombinationen ein Bild zu rekonstruieren.
- Die Größe des zu rekonstruierenden Bildes in Pixeln für jede Dimension.
- Die Größe des zu rekonstruierenden Volumens (Start- und Endpunkt eines Quaders innerhalb des USCT). Damit ist es möglich auch nur Teilbereiche des Zylinders zu rekonstruieren.
- Die Art der Bildrekonstruktion: Bei Amplitudenbildern werden alle absoluten Amplitudeninformationen eines A-Scans berücksichtigt, Phasenbilder beachten beim Addieren der Werte die Vorzeichen der Amplituden, Deconvolution-Bilder nutzen einen Entfaltungs-Filter, um die Anfänge der Signale zu detektieren und Envelope-Bilder bilden vor dem Addieren der Signale die Einhüllende.

- Die zu verwendenden Geometrie-Dateien. In ihnen sind die genauen Positionen der Sender und Empfänger im kartesischen Koordinatensystem gespeichert und werden für die Berechnung der Ellipsen und deren Position innerhalb des USCT zwingend benötigt.

Nach dem Starten der Rekonstruktion durchläuft die Software die Ordnerstruktur der Rohdaten, lädt dabei die gewünschten A-Scans und erzeugt nach dem genannten Prinzip anhand der Koordinaten des entsprechenden Senders und Empfängers nach und nach eine Matrix gemäß der Größenvorgabe des Bildes. Jedes Element der Matrix stellt ein Pixel des Bildes dar. In ihm werden die Werte der Amplituden aufaddiert, wodurch sich ein sehr großer Wertebereich ergibt.

Nach Fertigstellung wird die Matrix als Variable in einer MATLAB-Datei in einem angegebenen Verzeichnis auf der Festplatte abgespeichert. Die Datei enthält alle ausgewählten Bildmodi in einer separaten Variable, deren Name dem Bildmodus entspricht („amplitude“, „phase“, „deconv“ oder „envelope“). Nicht rekonstruierte Bildmodi werden durch leere Matrizen repräsentiert. Der Dateiname setzt sich aus dem Präfix „_oneImage_“ und der Größe in Pixeln zusammen, beispielsweise „_oneImage_x512_y512_z51.mat“.

Zusätzlich werden Informationen über das rekonstruierte Bild in einer separaten Datei abgespeichert. Deren Name setzt sich aus dem Präfix „_Info_“ und der Größe des Bildes in Pixeln zusammen, beispielsweise „_Info_x512_y512_z51.mat“. Als Informationen werden abgespeichert:

- die Flags-Variable, die zahlreiche Parameter beinhaltet, welche bei der Rekonstruktion verwendet werden
- der Start- und Endpunkt von Bildausschnitten der Rekonstruktion
- die physikalische Auflösung des Bildes in m/Pixel
- die verwendete Winkelkorrektur
- die Größe des Bildes in Pixeln
- die Parameter für die Signalentfaltung
- die Geometriedateien für Sender und Empfänger

4.2 MATLAB

MATLAB steht für „MATrix LABoratory“ und ist eine kommerzielle mathematische Software der Firma „The MathWorks Inc.“. Sie wird vor allem zur numerischen Lösung von mathematischen Problemen eingesetzt, bietet aber auch vielfältige Möglichkeiten zur grafischen Darstellung von Daten und Ergebnissen. Aktuell ist MATLAB in der Version 7.3. (auch als Release 2006b bezeichnet) erhältlich. Im IPE werden die Versionen 7.1 sowie 6.5 und 6.1 eingesetzt. Die Versionen sind weitestgehend abwärtskompatibel.

4.2.1 Programmierung in MATLAB

Die Programmierung erfolgt in einer proprietären und plattformunabhängigen Programmiersprache, die auf dem jeweiligen Rechner von MATLAB interpretiert wird. Ein Kompilieren und Linken wie in anderen Programmiersprachen ist daher nicht notwendig. Zur schnellen Ausführung von numerischen Berechnungen setzt MATLAB auf prozessoroptimierte Bibliotheken, bietet aber auch eine Just-in-time-Kompilierungstechnologie für einfache skalare Berechnungen an [The06].

Die Syntax ist sehr mathematisch orientiert und unterscheidet sich teilweise erheblich von der Hochsprachen-Syntax, wie sie beispielsweise C/C++ oder Java verwenden.

Ein großer Vorteil bei der Berechnung von mathematischen Problemen ist die durchweg vorhandene Unterstützung von Vektor- und Matrizen-Operationen, wodurch aufwendige Programmkonstrukte für elementweise Berechnungen nicht benötigt werden. Oft können ganze for-Schleifen entfallen und durch einzeilige Ausdrücke ersetzt werden. Die Deklaration von Variablen, die Angabe von Datentypen sowie die Zuordnung von Speicher entfällt bei Matlab gänzlich. Datentypen werden anhand des Wertes einer Variablen automatisch festgelegt, die Speicherzuweisung erfolgt dynamisch.

Als Datentypen kennt Matlab: [The00b]

- `char` - für Zeichenketten
- `NUMERIC` - unterteilt in die Typen `int8` (8-Bit Integer-Werte), `uint8` (8 Bit Integer-Werte ohne Vorzeichen), `int16`, `uint16`, `int32`, `uint32`, `single` (Gleitkommazahl mit geringer Präzision), `double` (Gleitkommazahl mit hoher Präzision) sowie `sparse` (speicherplatzsparend für Matrizen mit wenig belegten Elementen)
- `cell` - Zellen, die wiederum andere Datentypen enthalten können
- `structure` - Strukturen, die als Untermengen andere Datentypen enthalten.
- `function handle` - Referenzen auf MATLAB-Funktionen oder Benutzeroberflächen-Objekte.
- `User Class` - benutzerdefinierte Datentypen, die durch Matlab-Funktionen erzeugt werden

Quellcode wird in Dateien mit der Endung „.m“ als reiner ASCII-Text gespeichert. Er kann somit sowohl mit dem in MATLAB eingebauten Editor als auch jeglichen anderen ASCII-Editoren bearbeitet werden. Der MATLAB-Editor bietet darüber hinaus neben einem Syntaxhighlighting auch die Möglichkeit des Debuggings. Wie aus anderen Sprachen bekannt können Breakpoints gesetzt werden, an denen die Interpretation des Quelltextes während der Laufzeit unterbrochen wird.

Bei der Programmierung wird zwischen Skripten und Funktionen unterschieden. Während Skripte lediglich eine Art Stapelverarbeitung von Befehlen darstellen, können Funktionen Eingabe- und Ausgabeparameter erhalten. Eine MATLAB-Funktion hat folgende Syntax:

```
function y = average(x)
```

wobei `function` das Schlüsselwort des Befehls, `y` den Rückgabewert, `average` den Funktionsnamen und `x` einen Parameter darstellt.

Variablen die in einer Funktion definiert wurden, werden als lokale Variablen bezeichnet und sind außerhalb der betreffenden Funktion nicht sichtbar. Globale Variablen dagegen können auch über Funktionsgrenzen hinweg bekannt gemacht werden. Hierfür wird einer Variablen vor der ersten Verwendung in einer Funktion das Schlüsselwort `global` vorangestellt.

Die Namen der Quelltextdateien entsprechen im Allgemeinen dem Namen der primären Funktion. Darüber hinaus können innerhalb einer Datei beliebig viele Unterfunktionen angelegt werden, die nur für dateiinterne Funktionen sichtbar sind [The00b].

Neben den Dateien für die Speicherung von Quelltext bietet MATLAB die Möglichkeit Variablen in einem eigenen Datenformat persistent zu speichern. Details über das Datenformat finden sich in [The05].

Für eine ausführliche Erklärung der Funktionen von MATLAB sei auf das Hilfesystem sowie auf [The00b] verwiesen.

4.2.2 GUI-Gestaltung mit MATLAB

Eine Benutzeroberfläche in MATLAB besteht grundsätzlich aus drei Teilen:

- Komponenten wie z.B. Schaltflächen, Listen, Menüs, statische Texte usw. dienen der Interaktion mit dem Benutzer
- Fenster (in Matlab als Figure bezeichnet) beinhalten die Komponenten
- Callback-Funktionen, die durch Aktionen des Benutzers aufgerufen werden.

Während die Callback-Funktionen im Matlab-Editor als einfacher Quelltext implementiert werden, gibt es für das Layout einer Oberfläche zwei Möglichkeiten der Erstellung. Die erste Variante ist die Erzeugung von Benutzeroberflächen durch Quelltext. Dabei werden die Fenster und die Bedienelemente durch die Instanziierung von so genannten „User Interface Control Objects“ erzeugt. Über set- und get-Methoden können die Eigenschaften der Objekte angepasst werden.

Die zweite Möglichkeit bietet einen wesentlich höheren Komfort. Mit dem Layout-Editor GUIDE (Graphical User Interface Development Environment)

können Bedienelemente ausgewählt, hinzugefügt und angeordnet werden (Abbildung 4.10). Die Attribute der Elemente lassen sich über den „Property Inspector“ ändern. Die fertige Oberfläche wird als .fig-Datei gespeichert und gleichzeitig eine neue gleichnamige Quelltext-Datei mit den Funktionsrümpfen der Callback-Funktionen angelegt [AAA04].

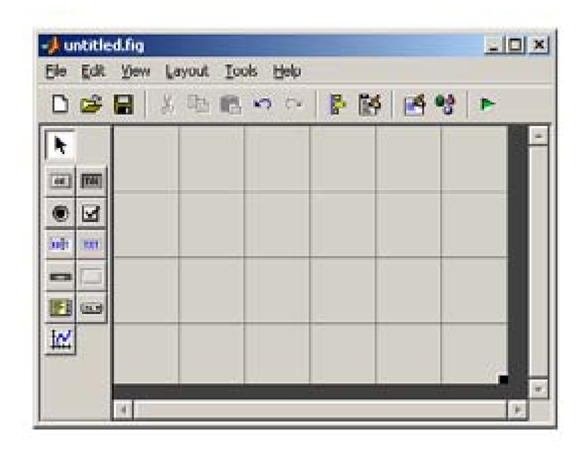


Abbildung 4.10: GUI-Layout mit GUIDE

Eine Benutzeroberfläche wird durch den Aufruf der jeweiligen MATLAB-Funktion gestartet. Diese lädt die .fig-Datei, die die Anordnung der Komponenten enthält. Die eigentliche Benutzerinteraktion wird durch die Callback-Funktionen der Bedienelemente implementiert. Neben dem eigentlichen Callback der bei einer Standard-Aktion (z.B. Klicken einer Schaltfläche, Bewegen einer Bildlaufleiste) ausgeführt wird, können zusätzlich Funktionen implementiert werden, die beim Erzeugen, Löschen oder dem Klicken auf ein Element ausgeführt werden. Für Fenster gibt es zusätzlich die Möglichkeit, Funktionen für das Drücken einer Taste auf der Tastatur, die Änderung der Fenstergröße, das Drücken der Schließen-Schaltfläche und das Ändern der Fensterposition zu hinterlegen.

Für alle Komponenten und Fenster können ferner Kontext-Menüs definiert werden, die bei einem Klick mit der rechten Maustaste ausgeführt werden. Als Menüleiste in einem Fenster kann sowohl die MATLAB-Standardleiste als auch eine selbst definierte Menüleiste mit beliebiger Anzahl Unterpunkte verwendet werden.

Des Weiteren bietet MATLAB einige vordefinierte Funktionen an, die einfache Dialoge öffnen und entsprechend der Dialogart Daten zurückliefern. Als Beispiel seien die häufig verwendeten Fehler- und Warndialoge, die einfachen Bestätigungs-Dialoge, sowie die Datei öffnen bzw. Speichern-unter-Dialoge genannt.

Bedienelemente stehen dem Software-Entwickler unter MATLAB 6.1 im Vergleich zu anderen Programmierumgebungen in recht geringer Anzahl zur Verfügung:

- Text-Felder - Anzeigen von statischem Text
- Edit Boxes - zum Einlesen und Ausgeben von Text
- Frames - Rahmen zur Untergliederung einer Benutzeroberfläche
- Pushbuttons - einfache Schaltflächen, die mit Text oder Grafiken belegt werden können.
- Togglebuttons - einrastfähige Schaltflächen
- Checkboxes - aktivier- und deaktivierbare Kästchen zur Auswahl mehrerer Optionen
- Radio Buttons - zur Auswahl exklusiver Optionen
- Popup-Menus - auch als Drop-Down-Menüs bezeichnet. Platzsparende Elemente zur Anzeige von Auswahllisten
- List Boxes - stellen eine Menge von Auswahlmöglichkeiten dar, aus denen der Benutzer eine oder mehrere markieren kann
- Sliders - Bildlaufleisten zur Auswahl von numerischen Werten aus einem vorgegebenen Bereich.

Eine besonders vielfältige Funktion hat außerdem das Axes-Objekt. Es kann jegliche von Matlab unterstützten Grafikdaten anzeigen. Darunter fällt unter anderem das Plotten von Daten sowie das Anzeigen von Bildern oder generierten Zeichnungen.

Genaue Beschreibungen und ausführliche Funktionsreferenzen finden sich im Hilfesystem von MATLAB sowie in [The02].

4.2.3 Vor- und Nachteile

Ein großer Vorteil von MATLAB ist die recht einfache Möglichkeit der Implementierung und Visualisierung von mathematischen Problemen. Die Software verfügt bereits über einen großen Funktionsumfang zur Datenanalyse, so. z.B. für Interpolation, Herausgreifen einzelner Datenabschnitte, Skalierung, Durchschnittswertberechnung, Auffinden von Maxima usw. Die Gestaltung von grafischen Benutzeroberflächen ist durch relativ wenig Aufwand möglich.

Des Weiteren ist die Plattformunabhängigkeit des Matlab-Codes ein großer Vorteil der Programmiersprache. In MATLAB implementierte Software kann auf allen Systemen mit einer MATLAB-Installation ausgeführt werden. Trotz der Skript-Sprachen-Eigenschaft werden Berechnung in ähnlicher Geschwindigkeit wie z.B. C++-Anwendungen ausgeführt [The06].

Als Nachteil ist zu nennen, dass Anwendungen lediglich durch die kommerzielle MATLAB-Software ausgeführt werden kann. Frei verfügbare MATLAB-Interpreter gibt es nicht. Abhilfe kann hier der Matlab-Compiler schaffen, der

aus MATLAB-Code C++-Code erzeugen kann. Somit entfällt die Interpretation durch MATLAB.

Ein weiterer Nachteil ist die nur eingeschränkt vorhandene Objektorientierung. In MATLAB erzeugter Code ist recht schwer wartbar. Dazu trägt auch die nicht notwendige Deklarierung von Variablen und Datentypen bei.

4.3 GUI-Design

Seit der steigenden Verbreitung von grafischen Benutzeroberflächen in den 1990er Jahren hat sich aus der Auseinandersetzung mit der Gestaltung dieser, eine eigene Wissenschaft gebildet. Die Software-Ergonomie ist heute ein sehr weites Feld. Deshalb soll an dieser Stelle keine detaillierte Betrachtung von Ergonomie-Erkenntnissen stehen, sondern lediglich auf die grundlegenden Prinzipien eingegangen werden, die bei der Gestaltung von grafischen Benutzeroberflächen beachtet werden sollten.

James Hobart stellte bereits im Jahr 1995 in [Hob95] zwölf Prinzipien für gutes GUI-Design auf. Trotz der rasanten Entwicklung in der Informatik seit dieser Zeit, sind diese noch heute aktuell.

Benutzer berücksichtigen Anwendungen müssen das Verhalten und die Perspektiven ihrer Benutzer berücksichtigen. Trotz der Individualität der Benutzer ergeben sich gemeinsame Charakteristiken wie z.B. im Bereich des Lernens und Wiedererkennens. Eine grafische Benutzeroberfläche sollte also beispielsweise bei einer erwarteten Eingabe dem Benutzer besser gewisse Vorgaben machen, als ihn völlig frei Werte wählen zu lassen.

Unterschiedliche Perspektiven beachten Was aus der Sicht des Entwicklers intuitiv und leicht verständlich ist, muss es noch lange nicht für den Benutzer sein. Dies sollte z.B. bei der Gestaltung von Icons, Tooltips, Textfeldern auf der Oberfläche, aber auch beim Benutzerhandbuch und der Dokumentation berücksichtigt werden.

Klare Gestaltung In eine ähnliche Richtung geht das Prinzip der klaren Gestaltung. Innerhalb einer Software sollte mit Begriffen konsistent umgegangen werden. Gerade bei mehreren Entwicklern kommt es oft vor, dass in einem Teil der Oberfläche bei ein und demselben Objekt von „Image“, in einem anderen von „Picture“ und einem dritten von „Figure“ gesprochen wird. Dies führt zu Unsicherheit beim Benutzer. Des Weiteren sollten einige reservierte Wörter und deren Bedeutung berücksichtigt werden (siehe Tabelle 4.1).

Konsistent gestalten Neben den Texten soll auch die gesamte Oberfläche konsistent im Verhalten gestaltet werden. Dadurch wird sich der Benutzer schneller zurechtfinden. Durch Berücksichtigung von Bedieneigenschaften

Text	Bedeutung und Verhalten	auf Buttons?	in Menüs?
OK	eingegebene Daten akzeptieren, bestätigen angezeigter Daten, anschließend das Fenster schließen	ja	nein
Cancel	Angezeigte Daten nicht bestätigen, Fenster schließen	ja	nein
Close	den aktuellen Prozess/Dialog schließen, zur Applikation zurückkehren, Ansicht von Daten schließen	ja	ja
Exit	Verlassen der Applikation	nein	ja
Help	Hilfesystem der Applikation aufrufen	ja	ja
Save	Speichern der Daten und im aktuellen Fenster bleiben	ja	ja
Save as	Speichern der Daten unter neuem Namen	nein	ja
Undo	Rückgängigmachen der letzten Aktion	nein	ja
Cut	markierte Buchstaben ausschneiden	nein	ja
Copy	kopieren der markierten Buchstaben	nein	ja
Paste	einfügen der kopierten oder ausgeschnittenen Buchstaben an der aktuellen Einfügeposition	nein	ja

Tabelle 4.1: Reservierte Wörter im GUI-Design. Quelle: [Hob95]

anderer erfolgreicher Software wird das Erlernen des Umgangs mit einer neuen Software erheblich erleichtert.

Visuelles Feedback Vor allem während Wartezeiten bei größeren Berechnungen ist es wichtig dem Benutzer Feedback über den Fortschritt zu geben. Als sehr geeignet gelten hier Fortschrittsbalken, im Besonderen mit einer Anzeige der verbleibenden Zeit. Ab welcher Zeit eine solche Anzeige notwendig wird, ist sehr variabel, als Daumenregel kann eine Dauer ab ca. vier Sekunden angegeben werden [Sko05].

Akkustisches Feedback Mit akkustischem Feedback sollte sehr sparsam umgegangen werden. Was zu Beginn der Benutzung einer Software noch ein nettes Feature sein mag, kann schon nach kurzer Zeit lästig werden. Deshalb ist es von Vorteil akkustische Signale nahezu ausschließlich für Fehlermeldungen zu verwenden und sonstige Töne lediglich optional anzubieten.

Texte verständlich halten Texte die auf der Oberfläche angezeigt werden, vor allem in Fehlermeldungen und Hinweisdialogen, sollten kurz und prägnant gehalten sein.

Nachvollziehbare Pfade Damit sich Benutzer in einer Anwendung nicht verirren, sollten zurückgelegte Wege nachvollziehbar sein. Hierfür sollte auf eine intuitive Menüstruktur mit möglichst wenigen Verschachtelungen gesetzt werden. Prägnant bezeichnete Titelleisten machen Dialoge leichter wiederauffindbar.

Tastaturunterstützung anbieten Trotz der Maus als primäres Interaktionsgerät, bevorzugen vor allem erfahrene Benutzer oft die Kombination aus Maus und Tastatur zur Bedienung einer Software. Zusätzlich bieten Tastaturkürzel effiziente Möglichkeiten häufig benutzte Funktionen sehr schnell aufzurufen. Dabei sollte auf Kombinationen aus maximal zwei gleichzeitig zu drückenden Tasten geachtet werden. Sehr beliebt ist auch die Belegung der Funktionstasten.

Konsistentes Look and Feel Das Aussehen und die Bedienbarkeit muss konsistent gehalten sein und sollte auch für den Aufbau von Schnittstellen berücksichtigt werden. Es sollten schon vor der Implementierung genaue Vorgaben festgelegt werden, an die sich die Programmierer eines Projektes halten müssen.

Modale und amodale Dialogfenster Bei der Implementierung von Dialogboxen ist darauf zu achten, dass sie entsprechend ihrer Absichten gestaltet werden. Speicher-unter-Dialoge beispielsweise sollten modal gestaltet werden, ebenso wie alle kritischen Dialoge. Amodale Dialoge hingegen können auch in den Hintergrund verschoben werden, während in der Hauptanwendung weitergearbeitet wird. Solche bieten sich vornehmlich für andauernde Prozesse wie beispielsweise Suchen, Auflistungen usw. an.

Bedienelemente gestalten Unter der großen Anzahl verschiedener benutzbarer Bedienelemente müssen für die jeweilige Aufgabe die passenden gefunden werden. Tabelle 4.2 gibt einen kurzen Überblick über die Verwendung von Bedienelementen

Des Weiteren sollte auch das Aussehen, die Bezeichnung und das Verhalten von Bedienelementen in der gesamten Software konsistent sein.

Bedienelement	**Anzahl der Auswahlmöglichkeiten**	**Art**
Menüleiste	max. 10 Einträge	statisch
Pull-Down-Menü	max. 12 Einträge	statisch
Kaskadierende Menüs	max. 5 Einträge pro Kaskade, 1 stufig kaskadierend	statisch
Pop-up-Menüs	max. 10 Einträge	statisch
Schaltfläche	1 pro Schaltfläche, max. 6 pro Dialog	statisch
Checkbox	1 pro Checkbox, max. 10-12 pro Gruppierung	statisches setzen / auswählen von Werten
Radiobutton	1 pro Radiobutton, max. 6 pro Gruppierung	statisches setzen / auswählen von Werten
Listenfeld	50 in der Liste, 8-10 davon angezeigt	dynamisches setzen / auswählen von (mehreren) Werten
Drop-Down-Listenfeld	20 in der Liste, immer nur 1 davon markiert	dynamisches setzen / auswählen von einzelnen Werten
Bildlaufleiste	dynamisch wählbar, abhängig von den angezeigten Daten	statisches setzen / auswählen aus einem Wertebereich

Tabelle 4.2: Eigenschaften von Bedienelementen im GUI-Design. Quelle: [Hob95]

Kapitel 5

Konzept und Design

5.1 Wahl der MATLAB-Version

In der USCT-Arbeitsgruppe werden unterschiedliche MATLAB-Versionen verwendet. Die erstellte Software soll dabei mit möglichst allen Versionen kompatibel sein. Aufgrund dessen soll zunächst diskutiert werden, unter welcher Version sie entwickelt wird.

Die MATLAB-Versionen 6.x stammen aus den Jahren bis 2002. Auf den Rechnern der Abteilung wird zumeist MATLAB 6.1 verwendet. Diese Version bietet eine recht eingeschränkte Möglichkeit zur GUI-Gestaltung. Allerdings verfügt das IPE über mehrere lizenzierte Toolboxen wie z.B. die Signalverarbeitungs-Toolbox, die für die Einbindung signalanalysierender Funktionen von großer Hilfe sein kann.

Matlab 7.1 dagegen bietet einige Überarbeitungen für die GUI-Gestaltung, wie z.B. das Gruppieren von Bedienelementen in Frames. Ebenso hat sich der eingeschlossene Funktionsumfang vergrößert. Durch einige Unterschiede in der Syntax jedoch, ist eine hundertprozentige Abwärtskompatibilität nicht gewährleistet. In der USCT-Gruppe sind deutlich weniger Lizenzen und keine Toolboxen für die Version 7.1 vorhanden.

Die Entscheidung fällt auf die Entwicklung unter MATLAB 6.1, da meist noch mit dieser Version gearbeitet wird und somit den Kompatibilitätsproblemen von MATLAB 7.x zu MATLAB 6.x aus dem Weg gegangen wird. Zusätzlich wird die Möglichkeit der Verwendung von Toolboxen unter Version 6.1 stärker gewichtet als die Erweiterungen in der GUI-Gestaltung in Version 7. Dennoch soll die Lauffähigkeit unter MATLAB 7 gewährleistet sein. Hierfür dienen ständige Tests und die Verwendung einer Syntax, die beide Interpreter-Sprachen teilen.

5.2 GUI-Konzept

Aufgrund der vielfältigen Aufgaben, die die Software erfüllen soll, ist es wichtig eine geeignete Strukturierung der Benutzeroberfläche zu finden, damit diese ihre Übersichtlichkeit behält und eine intuitive Bedienung ermöglicht.

Grundsätzlich lassen sich die Anforderungen in drei Gruppen unterteilen:

- Funktionen die auf den Rohdaten (A-Scans) einer USCT-Messung arbeiten
- Funktionen die auf den Daten rekonstruierter Volumen- oder Schichtbilder arbeiten
- Funktionen zur Visualisierung des USCT und Generierung von Parameterdateien für Messungen und Rekonstruktionen.

Ausgehend von den gruppierten Anforderungen können mehrere Wege zur Gestaltung einer Benutzeroberfläche beschritten werden.

Eine erste Möglichkeit ergibt sich durch die klassische Gestaltung einer Software mit einer alleinstehenden, alle Funktionen umfassenden Benutzeroberfläche wie in Abbildung 5.1 dargestellt. Die Funktionalität verbirgt sich hinter Schaltflächen, Menüs, Kontext-Menüs usw.

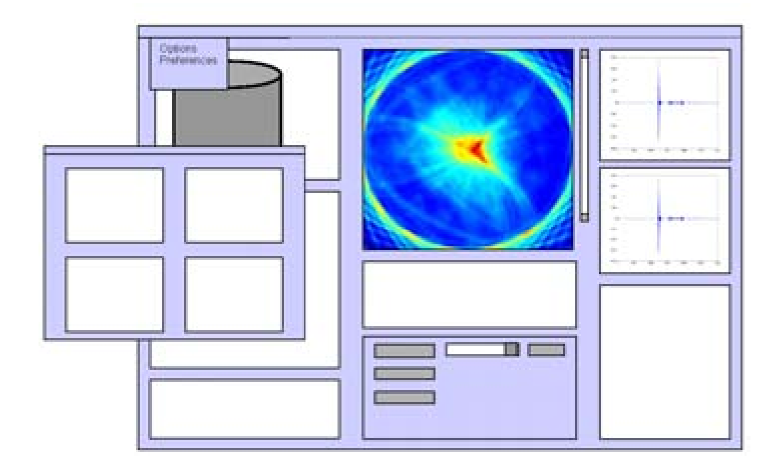


Abbildung 5.1: Entwurf einer Benutzeroberfläche: Alle Bedienelemente der Teilbereiche befinden sich in einem Fenster

Vorteil einer solchen Benutzeroberfläche ist die integrierte Verfügbarkeit aller Informationen. Dies lässt sich jedoch auch gleichzeitig als größter Nachteil auslegen: durch die zahlreichen gleichzeitig dargestellten Informationen kann die Verfolgung dieses Konzeptes schnell zur Unübersichtlichkeit führen. Dem Benutzer ist es nicht möglich, die für ihn irrelevanten Teilbereiche, Funktionen oder Informationen auszublenden.

Eine zweite Variante verschafft eine bessere Übersichtlichkeit durch die Ausblendung von Teilbereichen. Jede Gruppe erhält ihre eigene Benutzeroberfläche

(Abbildung 5.2). Zwischen den Gruppen kann über ein Registerkartensystem gewechselt werden. Auf jeder Registerkarte werden die entsprechenden Bedienelemente angezeigt.

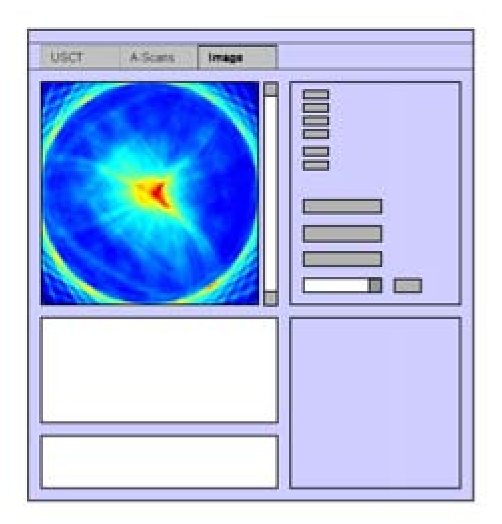


Abbildung 5.2: Entwurf einer Benutzeroberfläche: Trennung der Bereiche durch Registerkartensystem

Ein klarer Nachteil ergibt sich allerdings hier, da die Zusammenhänge zwischen den Gruppen schwer abgebildet werden können. Ein automatischer Wechsel wäre ebenso wenig benutzerfreundlich wie ein erforderlicher manueller Wechsel. Der Benutzer kann nie alle Informationen gleichzeitig erfassen. Des Weiteren ist eine Umsetzung mit den vorhandenen Bedienelementen als recht schwierig anzusehen.

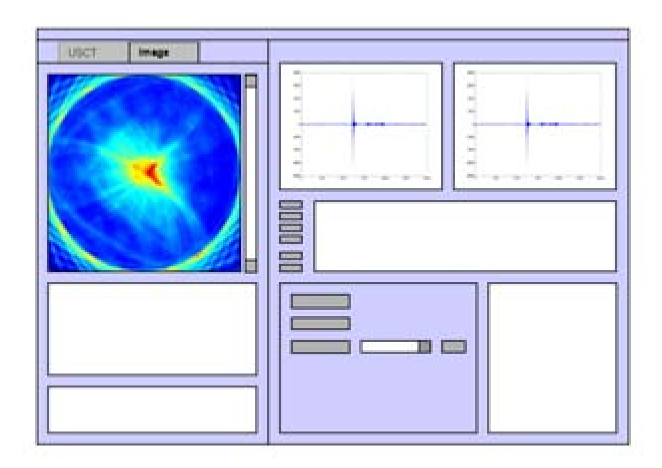


Abbildung 5.3: Entwurf einer geteilten Benutzeroberfläche: Zwischen Bild- und USCT-Bereich kann mit Registerkarten gewechselt werden

Eine hybride Version der bisher vorgestellten Konzepte ist in Abbildung 5.3 zu sehen. Da A-Scans sowohl mit der USCT-Visualisierung als auch mit rekonstruierten Bildern zusammenhängen, wird die Oberfläche zweiteilig erstellt. Dabei bleibt der A-Scan-Bereich ständig sichtbar, im zweiten Bereich kann zwischen der USCT-Visualisierung und den rekonstruierten Bildern gewechselt werden.

Vorteil dieser Oberfläche ist das Ausblenden von nicht gewünschten Informationen, wodurch eine bessere Übersichtlichkeit erreicht wird. Indes spricht die nicht vorhandene Möglichkeit der gleichzeitigen Informationserfassung von USCT- und Bilderbereich gegen diese Variante. Ebenso ist die Machbarkeit aufgrund der nicht vorhandenen Bedienelemente und der daraus resultierenden Notwendigkeit einer Emulation, ähnlich fraglich zu bewerten wie im vorherigen Vorschlag.

Durch eine dreiteilige Benutzeroberfläche wie sie in Abbildung 5.4 dargestellt wird, kann eine ähnliche Darbietung der Informationen wie in der ersten vorgestellten Variante erreicht werden. Darüber hinaus bietet sich dem Benutzer jedoch die Möglichkeit, nicht benötigte Teilbereiche zu schließen bzw. in die Taskleiste zu minimieren. Erreicht wird dies durch den Aufbau dreier eigenständiger Benutzeroberflächen, die dennoch miteinander interagieren können.

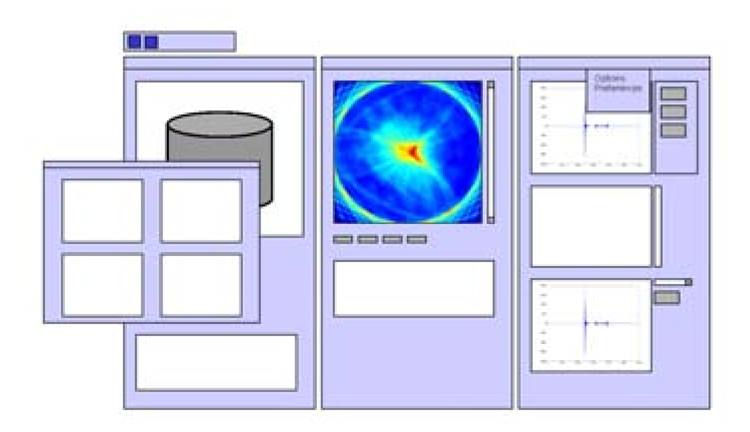


Abbildung 5.4: Konzept einer dreiteiligen Benutzeroberfläche mit eigenständigen Fenstern

Zur Ausrichtung der Fenster kann ein zusätzliches kleines Fenster dienen, das verschiedene Möglichkeiten zur Fensteranordnung vorgibt bzw. die Fenster beim Start der Software initial ausrichtet.

Nachteil dieses Konzeptes ist die Aufteilung der Funktionen auf mehrere MATLAB-Fenster, da bei der Ausführung von oberflächenübergreifenden Funktionen der Fokus zwischen den Fenstern gewechselt werden muss. Des Weiteren erhält jedes Fenster einen separaten Task, wodurch die Anwendung bei zusätzlichen Popup-Fenstern einen großen Raum in der Taskleiste benötigt. Die Erzeugung eines Hauptfensters mit mehreren Unterfenstern ist unter MATLAB nicht

möglich. Dadurch entsteht für den Benutzer ein erhöhter Verwaltungsaufwand, wenn beispielsweise alle drei Fenster verdeckt wurden und er sie durch Anklicken jedes einzelnen Fenster wieder in den Vordergrund des Desktops holen muss.

Entschieden wird sich trotz der genannten Nachteile für die zuletzt vorgestellte, dreiteilige, Variante, da sich dem Benutzer hierdurch eine große Variabilität bietet. Die leicht erhöhte Rechenzeit bei Fokuswechseln ist angesichts der zahlreichen weiteren Berechnungen, die durch die Software vorgenommen werden, sowie der Rechnerausstattung am Einsatzort vernachlässigbar.

Die Oberfläche teilt sich in drei Funktionsgruppen. Im A-Scan-Fenster werden alle relevanten Funktionen für die Anzeige und Bearbeitung von Rohdaten eingefügt. Dazu gehört unter anderem das Laden der A-Scans, die Anwendung von Transformationen usw. Zusätzliche Popup-Fenster werden für Informationen erzeugt, die nur auf Anfrage des Benutzers berechnet werden sollen. Zu nennen sind hier z.B. die platzraubende Anzeige mehrerer A-Scans, die Berechnung von A-Scan-Parametern oder die Anzeige von Informationen zum Experiment.

Im USCT-Fenster findet die Visualisierung des USCT-Messzylinders statt. Erreicht wird diese durch ein 3D-Modell der Sende- und Empfangselemente sowie eine 2D-Darstellung in der Draufsicht. Beide sind interaktiv zu bedienen. Es können Elemente als Sender bzw. Empfänger markiert werden. Darüber hinaus werden hier Funktionen zur Generierung von Parameterdateien aus den markierten Sendern und Empfängern für andere Softwaresysteme angeboten.

Das Rekonstruktionsbilder-Fenster bietet die Möglichkeit rekonstruierte Bilder des USCT zu laden, Bildbearbeitungsfunktionen darauf anzuwenden usw. Für die Darstellung von Volumenbildern dient ein Teilsystem, das Schichtbilder aus dem Volumen herausgreift. Über noch genauer zu spezifizierende Bedienelemente kann dann ein Volumen durchlaufen werden.

Zur besseren Übersicht sind nur die wichtigsten Funktionen und Informationen direkt auf der Oberfläche vorhanden, zusätzlich werden Kontext-Menüs und Funktions-Menüs angeboten.

5.3 Plugin- und Schnittstellen-Konzept

Eine wichtige Anforderung an die Software ist die Erweiterbarkeit. Diese wird in einem Plugin-Konzept umgesetzt, das an verschiedenen Stellen in der Software angeboten wird.

Grundidee hierfür ist das Auslesen bestimmter Ordner beim Start der Software, in denen die Quelltextdateien mit den entsprechenden Funktionen abgelegt werden können. Sie werden anschließend in Menüs oder Drop-Down-Feldern eingetragen, aus denen sie der Benutzer auswählt.

Es lassen sich zwei Plugin-Typen unterscheiden:

5.3.1 Eigenständige Plugins

Über eigenständige Plugins lassen sich beliebige Funktionen in die Software integrieren. Eine solche Variante wird in jedem der drei Teile der Benutzeroberfläche in der Menüleiste angeboten. In der Implementierung hat der Entwickler freie Hand, da er keine Übergabeparameter beachten muss. Die Funktionen sind eigenständig und werden durch die Software lediglich per Funktionsaufruf gestartet. Dies hat zur Folge, dass sich der Entwickler auch um eventuelle Darstellungen in eigenen Benutzeroberflächen kümmern muss.

Zur Einbindung in die Software muss die .m-Quelltextdatei mit einer gleichnamigen Funktion in das dafür vorgesehene Verzeichnis (siehe auch Abschnitt 5.4) gespeichert werden. In einer zusätzlichen, ebenfalls gleichnamigen, MATLAB-Datendatei muss eine String-Variable mit dem Text des Menüeintrages bereitgehalten werden.

Grundsätzlich ist es auch möglich, die vorhandenen Darstellungsformen der entwickelten Benutzeroberfläche, wie z.B. das Modell des Ultraschall-Computertomographen, zu benutzen. Der Zugriff darauf erfolgt über vorhandene Funktionen, wie sie in der Funktionsreferenz der Software dargestellt werden.

5.3.2 Plugins mit Parameterübergabe

Eine zweite Variante macht dem Entwickler strikte Vorgaben, die bei der Entwicklung von Plugins beachtet werden müssen. Im Gegenzug ist es mit ihnen jedoch sehr einfach die vorhandenen Eingabedaten und Visualisierungsmöglichkeiten zu nutzen. Diese Art von Plugins wird für die A-Scan-Transformationen und die Bildvergleichs-/vereinigungsfunktionen eingesetzt. Eine hybride Form der beiden Plugin-Arten wird für Operationen auf Bildern bzw. Bildausschnitten verwendet.

Der Aufbau unterscheidet sich von den eigenständigen Plugins vor allem im Rückgabewert. Dieser ist für Plugins dieser Art zwingend erforderlich, da die Daten nach Abarbeitung der Funktion weiterverarbeitet werden. Aufgerufen werden die Plugins mit der Übergabe von festgelegten Parametern, beispielsweise bei A-Scan-Transformationen die Daten eines A-Scans. Eine MATLAB-Datei mit dem Struct „inputParams“ als Inhalt spezifiert Art, Typ und Wert einer beliebigen Anzahl zusätzlicher Parameter, die der Funktion übergeben werden. Aufgrund der Verwendung eines Structs ändert sich die Anzahl der der Funktion übergebenen Parameter nicht, was einen einheitlichen Funktionsaufruf ermöglicht. Der Zugriff auf die im Struct enthaltenen Variablen ist dem Entwickler bekannt.

Eine Vereinfachung für Programmierer und Benutzer stellt die dynamische Generierung eines Dialoges anhand der Anzahl und Namen der Parameter dar. Sie bietet die Möglichkeit die numerischen Werte der Eingabevariablen „inputParams“ beim Aufruf des Plugins zu verändern.

Der Rückgabewert ist für jedes Plugin genau spezifiziert, so dass die Daten nach Abarbeitung weiterverwendet werden können. Im genannten Beispiel für

die A-Scan-Transformationen, werden die transformierten Daten beispielsweise geplottet.

5.4 Strukturierung der Software

Aufgrund der Skriptsprachen-Eigenschaften von MATLAB ist eine objektorientierte Programmierung nicht möglich, so dass zur Wahrung der Übersichtlichkeit eine Strukturierung der Software erfolgen soll.

Das Fehlen eines Package-Konzeptes, wie es beispielsweise Java anbietet, wird durch die Erzeugung einer Ordnerstruktur (Abbildung 5.5) kompensiert. Die Unterordner werden beim Starten der Software in die Pfadlisten der aktuellen MATLAB-Session eingetragen.

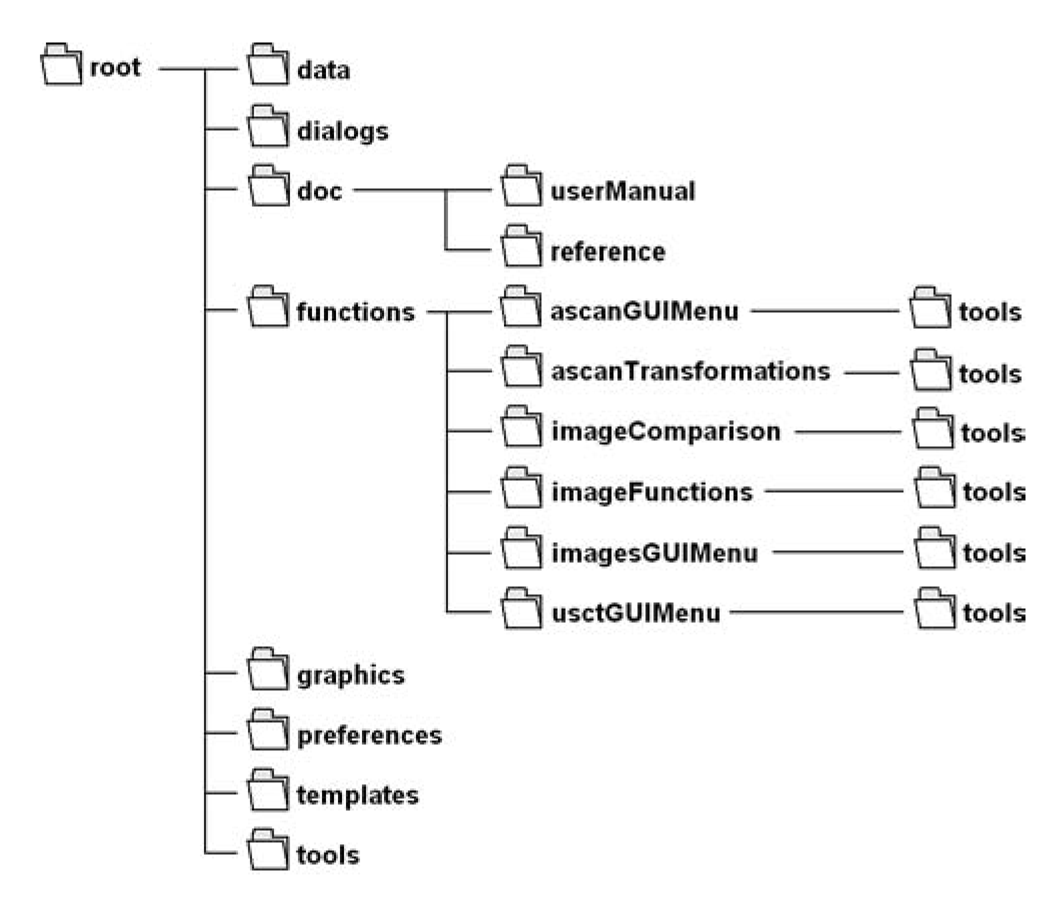


Abbildung 5.5: Ordnerstruktur der Software

Der Ordner in der ersten Hierarchieebene („root“) bezeichnet das Verzeichnis, in dem die Software abgelegt ist. In ihm befinden sich die ausführbaren Teile der Software, d.h. die drei Hauptfenster der Benutzeroberfläche sowie ein Start-Tool, das die Software lädt und die Fenster auf dem Bildschirm ausrichtet.

Im Unterordner „data“ werden MATLAB-Datendateien abgelegt, die für den Aufbau der Benutzeroberfläche benötigt werden. Dies sind vornehmlich die Geometrie-Dateien des USCT, die die Koordinaten der Sender und Empfänger liefern.

Der Ordner „dialogs“ auf der zweiten Hierarchieebene dient der Aufbewahrung der Dialoge, die die Software während der Laufzeit aufruft. Unter anderem ist hier die Anzeige von Informationen zu A-Scans und ein Dialog zum Plotten mehrerer A-Scans zu nennen.

Der Unterordner „doc“ beinhaltet die Dokumentation der Software. Er ist nochmals unterteilt in „userManual“- dem Benutzerhandbuch - und „reference“ - der Funktionsreferenz. Ersteres ist eine Möglichkeit für den Benutzer die Funktionalität der Software nachzuschlagen, Zweitere bietet Entwicklern eine Übersicht über den Quelltext und die Schnittstellen zwischen den Funktionen.

Sämtliche Plugins werden im Ordner „functions“ abgelegt. Dieser ist nochmals unterteilt in die Unterordner für die jeweiligen Plugin-Schnittstellen:

- „ascanGUIMenu“, „usctGUIMenu“ und „imagesGUIMenu“ für die Einbindung beliebiger Funktionen in der Menüleiste
- „ascanTransformations“ für die Einbindung von A-Scan-Transformationsfunktionen
- „imageComparison“ für die Einbindung von Bildvergleichs- und vereinigungsfunktionen
- „imageFunctions“ für die Einbindung beliebiger auf den Bilddaten arbeitenden Funktionen.

Jeder dieser Ordner besitzt einen zusätzlichen Unterordner „tools“, der zur Speicherung von Funktionen dient, die durch die Plugins aufgerufen werden.

„graphics“ enthält Grafikdateien, die in den Benutzeroberflächen verwendet werden. Diese werden z.B. für die Belegung von Schaltflächen benötigt.

Im Unterordner „preferences“ werden MATLAB-Dateien mit benutzerdefinierten Einstellungen gespeichert, die auch sessionübergreifend gespeichert werden sollen. Dazu zählen beispielsweise die Werte von Konstanten, sowie Informationen über das aktuelle Experiment auf dem gearbeitet wird.

Vorlagen für Funktionen, die die Plugin-Schnittstellen implementieren, sind im Ordner „templates“ abgelegt. Der Ordner „tools“ enthält alle weiteren Quelltextdateien mit Funktionen, die von der Software intern aufgerufen werden.

5.5 Datenhaltung und Datenaustausch

Innerhalb der Software ist es notwendig eine Datenhaltung und Möglichkeiten zum Datenaustausch zu finden, um von allen Stellen in der Software Zugriff auf Bedienelemente und benötigte Daten zu haben.

5.5.1 Zugriff auf Bedienelemente

Jede der drei Benutzeroberflächen arbeitet mit einer Variablen mit der Bezeichnung „handle“, die dem jeweiligen Fenster zugeordnet ist. In ihr sind Referenzen auf alle Bedienelemente der Oberfläche gespeichert. Zusätzlich können in ihr Daten gespeichert werden, die über die jeweiligen Callback-Funktionsgrenzen hinweg verfügbar sein sollen. Jede Callback-Funktion bekommt die Handles-Variable als Parameter übergeben.

Für den Zugriff auf die Bedienelemente werden primär die Referenzen der Handles-Variable verwendet, da die Interaktion zwischen den Bedienelementen größtenteils innerhalb einer Oberfläche stattfindet.

MATLAB bietet zudem mit den globalen Variablen (siehe Abschnitt 4.2.1) eine Möglichkeit von verschiedenen Funktionen aus auf gemeinsame Datenbereiche zuzugreifen. Interaktionen zwischen den Teilbereichen der Benutzeroberfläche werden daher über globale Variablen geregelt. Jede Oberfläche erhält eine solche und ist somit aus allen Bereichen der Software veränderbar. Die Referenzen auf die Bedienelemente werden nach Erzeugung der GUI an die globale Variable übergeben.

5.5.2 Datenzugriff

Der Zugriff auf Daten erfolgt analog zum Zugriff auf die Bedienelemente sowohl über die Handles-Variable, als auch über die globalen Variablen. Oberflächeninterne Informationen werden zwischen den Subfunktionen über die Handles-Variable ausgetauscht. Der Datenzugriff über die Oberflächengrenzen hinweg erfolgt über das Anhängen an das Struct der jeweiligen globalen Variable der Oberfläche.

Im Speziellen gibt es einige wichtige Informationen deren Datenhaltung hier näher erläutert werden soll:

Markierte Sender und Empfänger Informationen über markierte Sender und Empfänger werden zentral in der globalen Variable der USCT-Oberfläche gespeichert. Sie werden durch Matrizen folgender Form beschrieben:

$$Sender = \begin{bmatrix} x_{1,1} & \cdots & x_{1,24} \\ \vdots & \ddots & \vdots \\ x_{96,1} & \cdots & x_{96,24} \end{bmatrix} \tag{5.1}$$

wobei die Anzahl der Spalten der Anzahl der Senderlayer und die Anzahl der Zeilen der Anzahl der Sendernummern entspricht.

$$Empfaenger = \begin{bmatrix} x_{1,1} & \cdots & x_{1,48} \\ \vdots & \ddots & \vdots \\ x_{192,1} & \cdots & x_{192,48} \end{bmatrix} \tag{5.2}$$

wobei die Anzahl der Spalten der Anzahl der Empfängerlayer und die Anzahl der Zeilen der Anzahl der Empfängernummern entspricht.

Ist Sendernummer i aus Senderlayer j bzw. Empfängernummer i aus Empfängerlayer j markiert, so wird in der jeweiligen Matrix an der Stelle $x_{i,j}$ eine 1 eingetragen, andernfalls eine 0. Es werden Sparse-Matrizen verwendet, um den Speicherbedarf zu minimieren.

Angezeigter A-Scan Die Daten des aktuell angezeigten A-Scans werden in einer 1×3000 - Matrix abgelegt, so dass jedes Element einem Samplepunkt entspricht. Die Matrix ist durch die Speicherung in der globalen Variable der A-Scan-Oberfläche von allen Bereichen zugänglich. Analog ist die Nummer des entsprechenden Senderlayers, Sendeelements, Empfängerlayers und Empfängerelements des angezeigten A-Scans global verfügbar. Alle genannten Informationen werden als Unterdaten im Struct der globalen Variable der A-Scan-Oberfläche vorgehalten.

Angezeigtes rekonstruiertes Bild Ein rekonstruiertes Bild wird in einer Matrix mit einer Anzahl von Zeilen und Spalten entsprechend der Bildgröße in Pixeln gespeichert. Dabei werden sowohl die Daten des aktuellen Datensatzes (z.B. Volumenbild) in der globalen Variablen abgelegt wie auch das aktuell angezeigte Schnittbild, das in der Oberfläche für rekonstruierte Bilder dargestellt wird. Ebenso erfolgt die Speicherung von Bildinformationen im Struct der globalen Variable der Bilder-Oberfläche.

Benutzereinstellungen Im Gegensatz zu den bisher beschriebenen Daten müssen die Benutzereinstellungen persistent gespeichert werden, so dass eine Datei userPreferences.mat angelegt wird, in der sämtliche derartige Parameter vorgehalten werden. Diese sollen dann über einen Dialog editierbar sein.

Kapitel 6

Umsetzung

Nach der Vorstellung eines Grobkonzeptes für die Gestaltung der Benutzeroberfläche in Kapitel 5 soll nun schrittweise die Umsetzung der einzelnen Teilbereiche erläutert werden.

Entsprechend der Anforderungen kann bei der Gestaltung von Benutzeroberflächen davon ausgegangen werden, dass alle Arbeitsplätze in der Abteilung mit Bildschirmen und Rechnern ausgestattet sind, die eine Auflösung von mindestens 1152×864 Pixel bieten. Meist wird eine Bildschirmauflösung von 1280×1024 verwendet. Aufgrund dessen kann ein absolutes Layout der Bedienelemente mit entsprechenden Maßen gewählt werden. Unter Berücksichtigung der Bildschirmauflösungen wird für die drei Teiloberflächen jeweils eine Breite von 380 Pixeln verwendet. Als Höhe werden 850 Pixel gewählt.

6.1 A-Scan-Oberfläche

6.1.1 GUI

Die A-Scan-Oberfläche (Abbildung 6.1) unterteilt sich in einen Bereich zum Laden und Anzeigen der A-Scans, in dem zusätzlich alle relevanten damit verbundenen Operationen untergebracht werden, sowie einen Bereich für die A-Scan-Transformationen. Die Bereiche sind durch zwei schwarze Balken mit entsprechender Beschriftung kenntlich gemacht und dadurch optisch getrennt. Für die Anzeige von A-Scans und Transformationen werden die von MATLAB zur Verfügung gestellten Axes-Objekte (2, 9) verwendet.

Zur Auswahl von Senderlayer, Sendernummer, Empfängerlayer und Empfängernummer erhält die Oberfläche vier Listenfelder (5), in die die jeweils verfügbaren Zahlenwerte beginnend mit 0 eingetragen sind. Diese Objekte eignen sich hier besonders für den Einsatz, da sie nach den Prinzipien der GUI-Gestaltung am meisten Einträge aufnehmen können, während lediglich Teilmengen davon angezeigt werden. Die Felder werden so konfiguriert, dass eine oder mehrere Zeilen

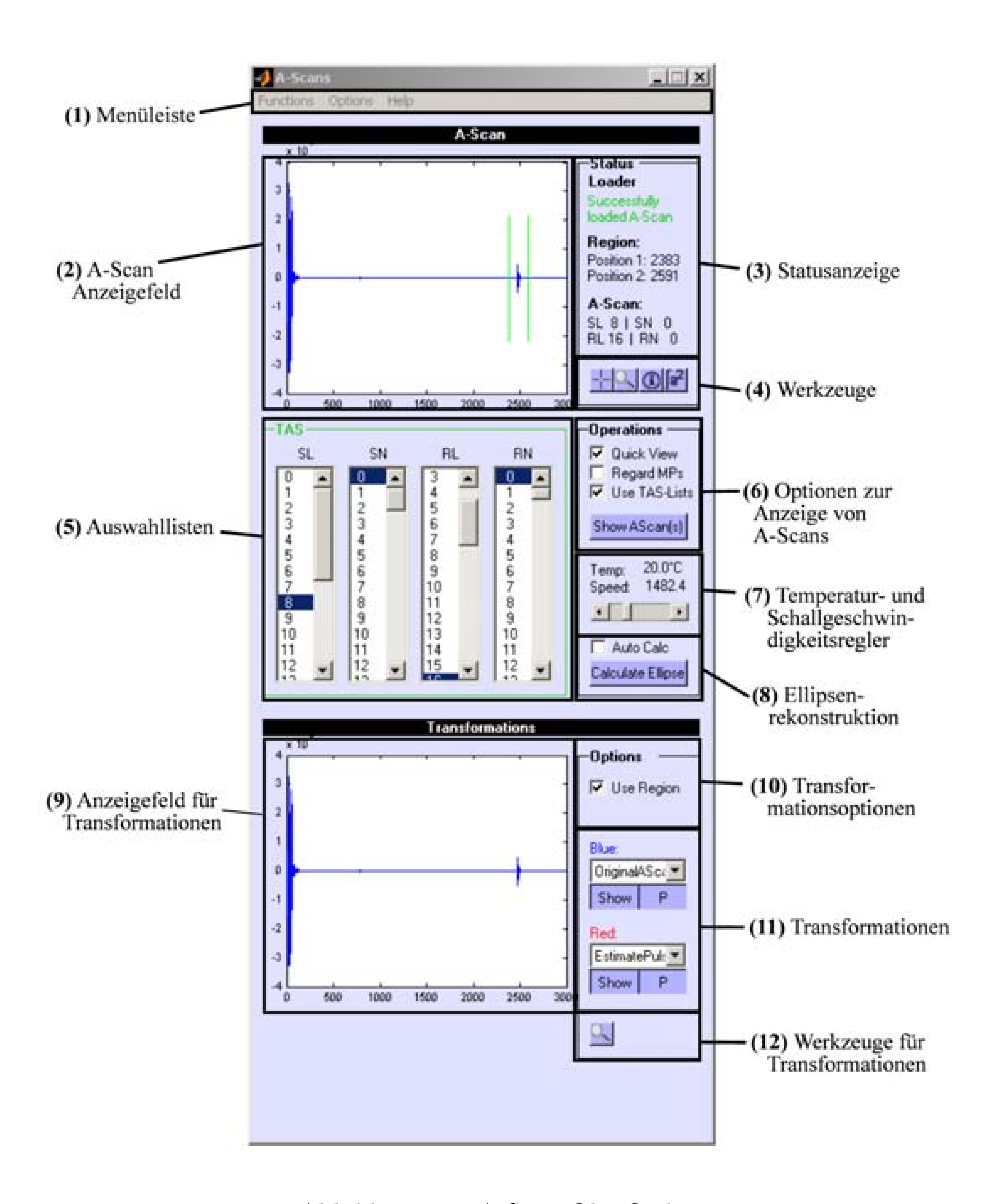


Abbildung 6.1: A-Scan-Oberfläche

darin markiert werden können.

Zusammengefasst in einem Rahmen stehen mehrere Statusanzeigen (3) zur Verfügung. Beim Laden eines A-Scans wird dort unter „Loader“ eine Erfolgs- (grüne Schriftfarbe) oder Fehlermeldung (rote Schriftfarbe) angezeigt. „Region“

stellt bei einer im A-Scan durchgeführten Markierung die untere und obere Grenze dieser numerisch dar. Unter „A-Scan“ wird die Sender-Empfänger-Kombination zum aktuell angezeigten A-Scan ausgegeben.

Informationen zum A-Scan und der dazugehörigen Messung erhält man durch Klicken auf eine Schaltfläche, die mit einem Infosymbol belegt ist. Diese ist gemeinsam mit den Schaltflächen zur Vergrößerung der Ansicht (Lupensymbol), zur Berechnung von A-Scan-Kennzahlen (Formelsymbol), sowie dem Werkzeug zur Auswahl eines Bereiches im A-Scan (Fadenkreuzsymbol) in einem Rahmen zusammengefasst (4).

Optionen zur Anzeige von A-Scans (6) werden zusammen mit einer Schaltfläche zum Laden und Anzeigen eines A-Scans in einen Bereich seitlich der Auswahllisten gepackt, um einen schnellen Zugriff und kurze Mauswege zu ermöglichen. Die Optionen werden durch Aktivieren bzw. Deaktivieren von Checkboxen ein- bzw. ausgeschaltet.

Ein weiterer Teil der Oberfläche ist ein Bedienelement zum Setzen der Temperatur des bei der Messung verwendeten Koppelmediums Wassers bzw. die daraus errechenbare Schallgeschwindigkeit (7). Abgebildet wird diese Funktion über eine Bildlaufleiste und zwei Textboxen, in denen die Auswahl und das Ergebnis der Geschwindigkeitsberechnung ausgegeben werden. Das Bedienelement kann damit auch gleichzeitig als Umrechner zwischen Wassertemperatur und Schallgeschwindigkeit benutzt werden.

Im Bereich der A-Scan-Transformationen dient erneut ein Axes-Objekt zur Anzeige der transformierten Daten (9). Unter „Options“ (10) dient eine Checkbox zum Aktivieren bzw. Deaktivieren der Transformationsoption zur Verwendung der markierten Region im A-Scan. Die Schaltfläche zur Anzeige einer Vergrößerung einer geplotteten Transformation ist, wie auch bei den A-Scan-Funktionen, in einem Rahmen (12) untergebracht.

Transformationen können im Anzeigefeld überlagert dargestellt werden. Hierfür werden zwei gleichartige Bedienelemente-Anordnungen verwendet (11), die beide ein Drop-Down-Menü zur Auswahl der Transformationsfunktion bieten. Über einen Togglebutton kann gewählt werden, ob die Eingabeparameter der Transformation vor der Ausführung geändert werden sollen. Ein zweiter Togglebutton zeigt die Transformation schließlich an bzw. löscht sie beim Deaktivieren des Buttons aus dem Anzeigefeld.

Die Menüleiste (1) komplettiert die A-Scan-Oberfläche. In ihr werden unter „Functions“ als Unterpunkte die über die Schnittstelle eingebundenen eigenständigen Funktionen angezeigt. „Options“ beinhaltet den Aufruf „Preferences“ zum Öffnen einer Oberfläche mit diversen Einstellungsmöglichkeiten der Software. „Help“ bietet einen Link zum Öffnen des User Manuals.

6.1.2 Funktionalität

Beim Starten der A-Scan-Oberfläche werden nach der Initialisierung der Bedienelemente zunächst einige grundlegende Einstellungen vorgenommen. Mit dem MATLAB-Befehl „addpath“ werden alle notwendigen Unterverzeichnisse der Software eingebunden. Anschließend liest die Software die Datei „UserPreferences.mat“ aus dem Ordner „preferences“ aus, in der die benutzerdefinierten Einstellungen gespeichert sind. Daraus leitet sich das Verzeichnis der Messung ab, auf der im Folgenden gearbeitet wird.

Die Umsetzung der Schnittstellen beginnt mit dem Auslesen der entsprechenden Unterordner der Software. Für die A-Scan-Transformationen werden die erhaltenen Informationen über die Funktionsnamen direkt in das Drop-Down-Menü geschrieben, während bei den eigenständigen Funktionen in der Menüleiste zunächst nach der, dem Dateinamen entsprechenden, .mat-Datei gesucht wird, um das darin enthaltene Label für den Eintrag in das Menü zu erhalten.

Die grundlegende Funktion der A-Scan-Oberfläche ist die Auswahl von A-Scans aus dem aktuellen Experiment anhand eines ausgewählten Senders und Empfängers. Diese erfolgt durch die Markierung einer oder mehrerer Zeilen aus den Listenfeldern. In den zugehörigen Callback-Funktionen werden die Indizes der Markierung ausgelesen. Anhand dieser kann im Inhalt des Listenfeldes der Wert an dieser Stelle ermittelt werden. Die Information über die Markierung wird jeweils in einer globalen Variable gespeichert. Anschließend werden die globalen Matrizen zur Speicherung der markierten Sender und Empfänger (siehe 5.5.2) aktualisiert.

Die Schaltfläche „Show AScan(s)“ überprüft in ihrer Callback-Funktion zunächst wie viele Sender-Empfänger-Kombinationen in den Listenfeldern ausgewählt sind und leitet entsprechend weitere Schritte ein.

Ist in jedem Listenfeld nur ein Element ausgewählt, wird die Subfunktion „loadAndShowAScan“ aufgerufen, die anhand der gespeicherten Information die Daten des A-Scans zur derzeit ausgewählten Sender-Empfänger-Kombination aus dem eingestellten Experimentverzeichnis lädt. In der Funktion „normalize“ wird der Offset des A-Scans durch Subtrahieren des Mittelwertes entfernt und die Daten in die Form eines Zeilenvektors umgewandelt. Schließlich plottet die Funktion die Daten in das Anzeigefeld und aktualisiert die Statusinformationen.

Bei der Auswahl mehrerer Sender-Empfänger-Kombinationen werden die Markierungen in for-Schleifen durchlaufen und in einer tabellarischen Struktur aufgelistet:

Senderlayer	**Sendernumber**	**Receiverlayer**	**Receivernumber**
:	:	:	:

Tabelle 6.1: Datenstruktur mehrerer Sender-Empfänger-Kombinationen

Diese n-zeilige Liste wird der Subfunktion „loadMultipleAScans“ übergeben,

die die Daten in einer Matrix mit 3000 Spalten und n Zeilen zurückliefert. Dadurch kann eine Zuordnung zwischen Sender-/Empfängernummer und Daten über den Zeilenindex erfolgen.

Innerhalb des Dialoges „multipleAScanPlot“ werden anschließend über eine globale Variable die Daten gesetzt und ein neues Fenster geöffnet, in dem die gewählten A-Scans geplottet dargestellt werden (Abbildung 6.2). Zum Aufbau dieses Fensters wird auf die Funktion „Scrolling Figure Demo“[1] von *Evan Brooks* zurückgegriffen, dessen Quelltext auf die Bedürfnisse der Software angepasst wurde.

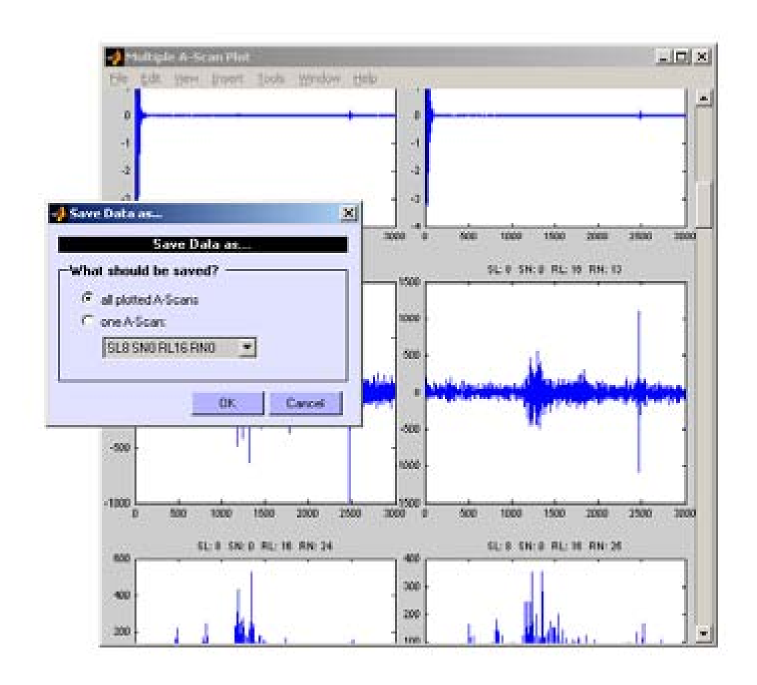


Abbildung 6.2: Oberfläche zur Anzeige mehrerer A-Scans und Oberfläche zum Abspeichern der A-Scan-Daten

Die Daten der angezeigten A-Scans können im MATLAB-Datenformat abgespeichert werden. Hierfür wird jedem Plot ein Kontextmenü zugeordnet, das in einem Unterpunkt den Befehl „Save Data as...“ als Auswahlmöglichkeit bietet. Die zugehörige Callback-Funktion öffnet einen Dialog in dem über Radiobuttons ausgewählt werden kann, ob alle oder der einzelne A-Scan gespeichert werden soll. Im zweiten Fall werden die Daten in eine Datei mit auswählbarem Namen exportiert, im ersten Fall erfolgt eine Art Stapel-Abspeicherung, bei der die Sender-/Empfänger-Nummern zur Namensgenerierung der Dateien dienen.

[1]Demo eines Fensters zum Plotten mehrerer Datensätze, das mit einer Bildlaufleiste versehen ist. Quelle: MATLAB File Exchange, `http://www.mathworks.com/matlabcentral/fileexchange`, Stand: 28.08.2006

Für die Auswahllisten zur Sender- und Empfängerauswahl können verschiedene zusätzliche Optionen gewählt werden:

Quick View Die Quick-View-Option ermöglicht es, A-Scans sofort nach der Auswahl in den Listenfeldern zu laden und anzuzeigen. Ein Drücken des Laden-Buttons entfällt und bietet so eine Möglichkeit schneller durch Datensätze zu navigieren. Die Funktion kann über eine Checkbox aktiviert bzw. deaktiviert werden. Dahinter verbirgt sich ein Flag, das bei der Aktivierung gesetzt wird. In den Callback-Funktionen der Listenfelder wird dann die „loadAndShow“-Funktion aufgerufen, die auch der „Load A-Scan(s)“-Button benutzt. Zusätzlich wird die zentrale Datenhaltung sowie die 3D und 2D-Darstellung aktualisiert.

Motorpositionen berücksichtigen Da an einer Motorposition für einen Sender nur bestimmte Empfänger die Daten aufnehmen, sind für einige in den A-Scan-Listen angebotenen Sender-Empfänger-Kombinationen keine Daten verfügbar. In der Checkbox „Regard MPs“ wird die Option des Ausblendens von Empfängern, die für die aktuell ausgewählten Sender nicht vorkommen können, angeboten. Ist die Option aktiviert, werden je nach ausgewählten Sendernummern die zugehörigen Empfängernummern auf der gleichen Motorposition berechnet und in die Liste eingetragen. Dies geschieht in der Callback-Funktion der Sendernummer-Liste sowie nach Aktivierung der Checkbox:

- Markierte Sendernummer(n) auslesen und mit dem Modulo-Operator auf die 6 verfügbaren Motorpositionen umrechnen

$$motorposition = markedSN \bmod 6 \tag{6.1}$$

- Sortieren der Motorpositionen
- Durchlaufen aller Sendernummern und Berechnung der dazugehörigen Empfängernummern:

$$RN = \begin{cases} (motorposition \cdot 2) + [0, 12, 24, ..., 180] \\ (motorposition \cdot 2) + [0, 12, 24, ..., 180] + 1 \end{cases} \tag{6.2}$$

- Aktualisieren des Empfängernummern-Listenfeldes
- Aktualisieren der Markierung innerhalb des Listenfeldes. Dabei werden nicht mehr angezeigte, zuvor ausgewählte, Empfänger aus der Auswahl gestrichen.

Rahmen (4) in der Abbildung 6.1 zeigt verschiedene Werkzeuge zur Anwendung auf A-Scans.

Die Fadenkreuz-Schaltfläche dient der Auswahl eines Bereichs im A-Scan, auf den weitere Aktionen eingeschränkt werden können. Die Selektion erfolgt mit dem MATLAB-Befehl

```
values = ginput(2);
```

Dieser liefert zwei Punkte jeweils mit x- und y-Koordinate in der aktuellen Skalierung der Achsen. D.h. bei einem A-Scan, der auf der x-Achse 3000 Samplepunkte aufträgt, liefert diese Auswahl x-Werte zwischen 1 und 3000. Anhand der x-Koordinate wird im vorliegenden Fall ein Bereich eingeschränkt. Eine Überprüfung der Markierungsreihenfolge findet statt, indem ein Vergleich der beiden Werte durchgeführt und diese gegebenenfalls getauscht werden, so dass die erste Komponente einer Markierung immer niedriger ist als die zweite. Die Markierung wird in der globalen Variable der A-Scan-Oberfläche gespeichert. Zur Visualierung werden zwei Linien an den entsprechenden Stellen in den A-Scan-Plot eingezeichnet (Abbildung 6.3).

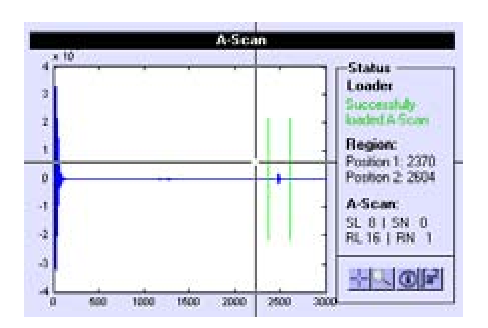


Abbildung 6.3: Auswahl eines Bereichs in einem A-Scan mit der Funktion ginput. In grün werden die Grenzen der Markierung eingezeichnet.

Die Callback-Funktion der Lupen-Schaltfläche öffnet ein neues Fenster (Abbildung 6.4), in dem ein gegenüber der A-Scan-Oberfläche vergrößertes Axes-Objekt das Plotten der Daten übernimmt. Dem Fenster werden mit einer globalen Variablen die Daten des aktuell in der A-Scan-Oberfläche angezeigten A-Scans übergeben. Zusätzlich erhält es die zugehörige Sender-Empfänger-Kombination, die dann in Textfeldern angezeigt wird. Auf das Axes-Objekt ist ein Kontextmenü definiert, in dem jeweils eine Funktion zum Exportieren des Plots in das JPEG und PNG-Format angeboten wird. Die Funktionen rufen zunächst einen Speichern-unter-Dialog auf, durch den der Dateipfad gewählt werden kann. Anschließend wird mit dem MATLAB-Befehl

```
saveas(handles.ascanEnlargedAxes,outfile);
```

das Fenster in eine Grafik umgewandelt, anhand der gewählten Dateiendung das Format gewählt und schließlich an der angegebenen Stelle auf der Festplatte gespeichert.

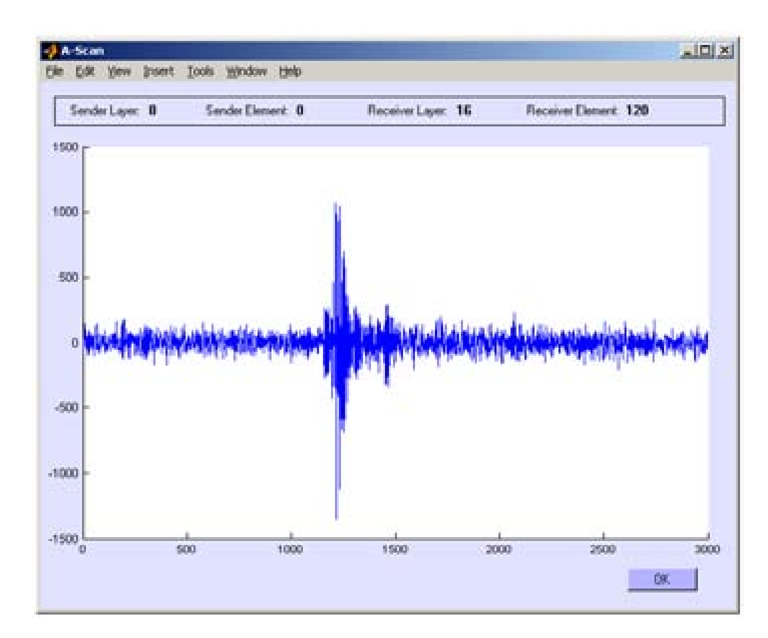


Abbildung 6.4: Oberfläche zur vergrößerten Ansicht eines A-Scans.

Die Info-Schaltfläche öffnet ein Popup-Fenster, in dem zum aktuell geladenen A-Scan Informationen und Eigenschaften anzeigt werden (Abbildung 6.5). Diese erhält die Software größtenteils durch das Laden der Datei „info.mat", die bei jeder Messung mit dem USCT angelegt wird (siehe auch Abschnitt 4.1.3). Des Weiteren werden Informationen bezüglich der Datenstruktur beim Laden eines A-Scans ermittelt.

Einige Kennzahlen eines A-Scans werden im Dialog „AScanProperties" berechnet und ausgegeben (Abbildung 6.6). Für die Berechnung werden größtenteils MATLAB-eigene Funktionen verwendet. Deren Vorgehensweise sei im Folgenden kurz mathematisch dargestellt. X bezeichnet dabei die Menge der n Messwerte x_i.

- Durchschnittliche Amplitude (mean)

$$mean(X) = \frac{\sum_{i}^{n} x_i}{n} \tag{6.3}$$

- Durchschnittliche Amplitude (median) mit X als aufsteigend oder absteigend geordnete Menge

$$median(X) = \begin{cases} x_{(n+1)/2} & \text{n ungerade} \\ \frac{1}{2}(x_{(n/2)} + x_{(n/2+1)}) & \text{n gerade} \end{cases} \tag{6.4}$$



Abbildung 6.5: Dialog zur Anzeige von Informationen zu einem A-Scan und dem dazugehörigen Experiment.

- Varianz

$$var(X) = \cdot \sum_{i}^{n}((x_i - mean(X))^2 \frac{1}{n}) \tag{6.5}$$

- Standardabweichung

$$std(X) = \sqrt{var(X)} \tag{6.6}$$

- Maximale Amplitude und Stelle an der diese auftritt

$$max(X) = \{x_i | x_i > x_j \forall x_j\} \text{ an der Stelle } i \tag{6.7}$$

- Minimale Amplitude und Stelle an der diese auftritt

$$min(X) = \{x_i | x_i < x_j \forall x_j\} \text{ an der Stelle } i \tag{6.8}$$

- Signal-Rausch-Verhältnis

$$snr(X)|_{dB} = 10 \cdot log\left(\frac{max(X)}{median(X^2)}\right) \tag{6.9}$$

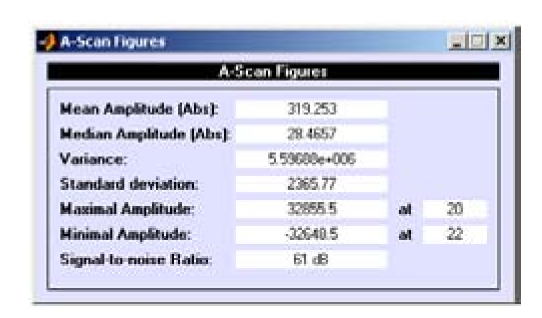


Abbildung 6.6: Dialog zur Anzeige von Kennzahlen eines A-Scans

Die Funktionen zur vergrößerten Anzeige des A-Scans, zum Öffnen des Info-Dialogs sowie zum Öffnen des Kennzahlen-Dialoges sind ebenfalls über ein Kontext-Menü zugänglich, das dem A-Scan-Anzeige-Feld zugeordnet ist.

Hinter der Checkbox „Use TAS-Lists“ verbirgt sich ein Flag, das gesetzt wird, um ausdrücklich die Markierungen in den Auswahllisten der A-Scan-Oberflächen zum Laden der Daten zu verwenden. Ist diese Box hingegen deaktiviert, wird das Listenfeld „Custom Sender-Receiver-Combinations“ (siehe auch 6.2.2) in der USCT-Oberfläche verwendet.

Der Bereich zur Einstellung der Wassertemperatur beruht auf einer Bildlaufleiste, deren Wertebereich initial auf das Intervall [15, ... ,35] eingestellt ist. Die bei einer Bewegung des Reglers ausgeführte Callback-Funktion liest den aktuellen Wert des Objektes und trägt ihn in das darüber liegende Textfeld mit dem Zusatz „°C“ ein. Anschließend wird die Ausbreitungsgeschwindigkeit des Ultraschalls im Medium Wasser bei der gewählten Temperatur mit der Funktion „soundSpeed“ berechnet, die aus der Rekonstruktionssoftware entnommen wurde. Diese nutzt zur Bestimmung der Schallgeschwindigkeit die Formel nach *Marczak* [Nat04]. Der errechnete Wert wird in das zusätzliche Textfeld eingetragen.

Auf die Berechnung der Rekonstruktionsellipsen wird in Abschnitt 6.4.4 näher eingegangen.

Der Bereich „Transformations“ basiert auf den Daten des A-Scans, der derzeit in der A-Scan-Oberfläche geladen ist und angezeigt wird.

Die Callback-Funktion der Drop-Down-Menüs liest die Namen der angebotenen Funktionen sowie die aktuelle Markierung aus. Der Name wird anschließend in der Handles-Variable gespeichert und repräsentiert nun, welche Funktion als Transformation aufgerufen werden soll.

Mit den beiden Parameter-Togglebuttons wird jeweils ein Flag gesetzt, das angibt, ob der Benutzer evtl. vorhandene zusätzliche Parameter vor der Ausführung der Transformation einsehen oder ändern möchte.

Der Aufruf der Transformation erfolgt durch Betätigen des „Calculate“-Togglebuttons. Mittels des Namens der Transformation wird zunächst nach einer MATLAB-Datei dieser Bezeichnung gesucht, in der zusätzliche Parameter gespeichert sind. In Abhängigkeit von der Existenz dieser Datei (und damit der

Existenz zusätzlicher Parameter) sowie des Benutzerwunsches nach einer Änderung der Parameter, wird die entsprechende MATLAB-Datei geladen und die Funktion „readInParameters“ mit deren Werten aufgerufen.

Die Funktion „readInParameters“ erzeugt anhand der Anzahl und Namen der in der MATLAB-Datei gespeicherten Variablen einen modalen Dialog (Abbildung 6.7). Die Höhe des Fensters berechnet sich aus der Anzahl der Parameter. Ebenso werden anhand der Anzahl der Parameter statische Textfelder mit den Namen der Variablen erzeugt. In Edit-Feldern werden die Werte der Variablen eingetragen, und können darin verändert werden. Mit Betätigen des OK-Buttons schreibt die Funktion die veränderten Parameter zurück in die Datei.

Abbildung 6.7: Dynamisch generierter Dialog zum Ändern von Parameterwerten.

Anschließend wird die Transformationsfunktion mit den gewählten Parametern aufgerufen. Das Ergebnis jeder Berechnung ist ein Datenarray, welches in das Transformations-Anzeige-Feld geplottet und in einer globalen Variable gespeichert wird. Wird der Togglebutton deaktiviert, entfernt die Funktion den jeweiligen Plot. Durch die doppelte Verfügbarkeit der Funktionalität, können Daten zweier Transformationen in unterschiedlichen Farben in das Feld geplottet werden.

Ein für das Anzeigefeld definiertes Kontextmenü bietet die Möglichkeit die Daten beider Transformationen in Dateien im MATLAB-Datenformat zu schreiben. Als Quelle wird die globale Variable der Transformationsdaten verwendet, deren Inhalt in eine Variable des Namens „data“ kopiert wird und mit einem vom Benutzer über einen Speichern-unter-Dialog gewählten Namen abgespeichert wird.

Die Lupen-Schaltfläche öffnet den selben Dialog zur vergrößerten Ansicht eines Plots wie auch bei der Vergrößerung von A-Scans. Zuvor werden die Daten aus der

globalen Variable mit den transformierten Daten gefüllt. Durch die Verwendung des identischen Dialoges, bleibt auch die Funktionalität zum Exportieren der geplotteten Daten in die Grafikformate JPEG und PNG erhalten.

Optional kann über die Checkbox „Use Region“ ausgewählt werden, ob eine Transformation als Arbeitsdaten lediglich den gewählten Ausschnitt aus den A-Scan-Daten erhält. Dabei werden die Arrayelemente zwischen den im A-Scan-Anzeigefeld gewählten Grenzen aus dem A-Scan herausgegriffen und als Übergabeparameter an die Transformationsfunktion gespeichert.

Die A-Scan-Oberfläche wurde aufgrund ihrer Wichtigkeit für die Software zur Unterbringung des Dialoges für benutzerdefinierte Einstellungen genutzt. Über die Menüleiste kann die Funktion „preferencesGUI“ aufgerufen werden (Abbildung 6.8). Diese liest die Werte aus der Datei „userPreferences.mat“ im Unterordner „preferences“ aus und stellt dem Benutzer den Inhalt grafisch dar. Unter anderem kann hiermit das aktuelle Experimentverzeichnis ausgewählt werden. Benutzt wird hierfür die vom MATLAB File Exchange genommene Funktion „uigetfolder“[2] von *Neil Rutland*, um die grafische Auswahl eines Verzeichnisses zu ermöglichen.

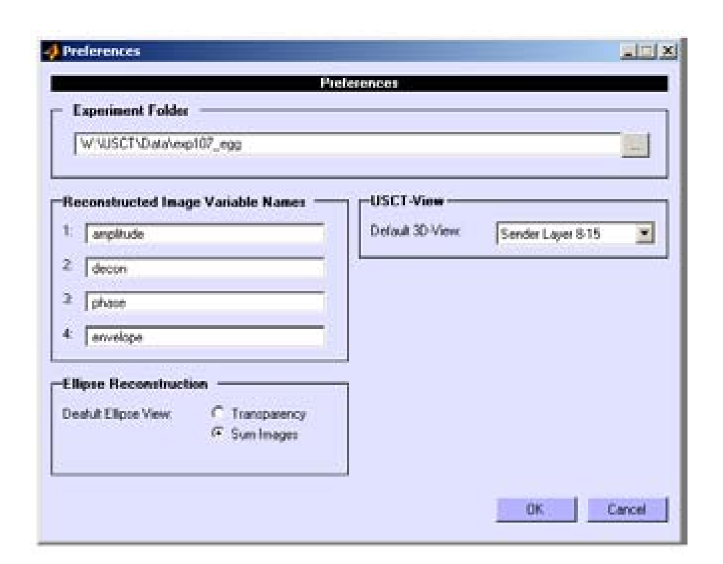


Abbildung 6.8: Oberfläche zur Änderung der benutzerdefinierten Einstellungen

Durch Betätigen des OK-Buttons setzt die Funktion die globalen Variablen neu und speichert die Einstellungen zurück in die Datei „userPreferences.mat“.

Wie auch die anderen beiden Hauptoberflächen besitzt die A-Scan-Oberfläche eine Schnittstelle für beliebige Erweiterungen. Bei der Initialisierung der Oberfläche wird - wie zu Beginn des Kapitels beschrieben - der entsprechende Ordner

[2] Funktion zum Öffnen eines Dialoges, mit dem ein Ordner auf der Festplatte ausgewählt werden kann. Quelle: MATLAB File Exchange, `http://www.mathworks.com/matlabcentral/fileexchange`, Stand: 30.08.2006

ausgelesen und zugehörige MATLAB-Dateien gesucht, in denen ein beliebiger Name für die Funktion angegeben werden kann. Dieser wird in das Menü eingetragen, während als Callback-Funktion der eigentliche Funktionsname gespeichert wird. Durch einen Klick auf einen Eintrag kann dann die jeweilige Funktion aufgerufen werden.

6.2 USCT-Oberfläche

6.2.1 GUI

Die USCT-Oberfläche (Abbildung 6.9) dient der Visualisierung des realen Ultraschall-Computertomographen. Als Grundelement hierfür dient ein dynamisch aus den Geometrie-Dateien generiertes 3D-Modell (2), das alle Sendeelemente des USCT darstellt. Jedes Sendeelement repräsentiert zusätzlich vier Empfängerelemente, die sich aus der geometrischen Anordnung der Ultraschallwandler in einem TAS ergeben. Die einzelnen Elemente können mit der rechten Maustaste angeklickt werden, um sie über ein Kontextmenü als Sender bzw. Empfänger zu markieren. Die Einfärbung der Elemente verrät den aktuellen Markierungszustand:

- **blau:** Grundfarbe der Elemente. Das Element ist nicht markiert.
- **rot:** Das Element ist als Sender markiert
- **grün:** Ein oder mehrere der - dem Element zugeordneten - Empfänger sind markiert
- **gelb:** Das Element ist als Sender markiert und gleichzeitig sind zugeordnete Empfänger markiert.

In einem Rahmen (3) sind Ansichtsoptionen untergebracht. Zwei Schaltflächen dienen zum Drehen des USCT-Modells im bzw. gegen den Uhrzeigersinn. Das Modell kann in drei Blöcke unterteilt werden, die über das Drop-Down-Menü ausgewählt werden können. Ein Togglebutton aktiviert bzw. deaktiviert die 3D-Rotationsfunktion. Mit dieser kann das Modell über die Maussteuerung in alle beliebige Richtungen frei gedreht werden.

Die 2D-Ansicht (4) zeigt den Ultraschall-Computertomographen aus der Draufsicht. Dabei wird durch die Anzeige jeweils ein Senderlayer repräsentiert. Eine zum 3D-Modell analoge Interaktion ermöglicht das Markieren der Elemente als Sender und Empfänger.

Die Kontrollfunktionen für die 2D-Darstellung werden im Rahmen „2D-Control“ (5) geboten. Die Motorposition kann über zwei Schaltflächen inkrementiert bzw. dekrementiert werden. Ebenso ist es möglich diese direkt in die Editbox einzutragen. Zur Visualisierung der geänderten Motorposition wird das

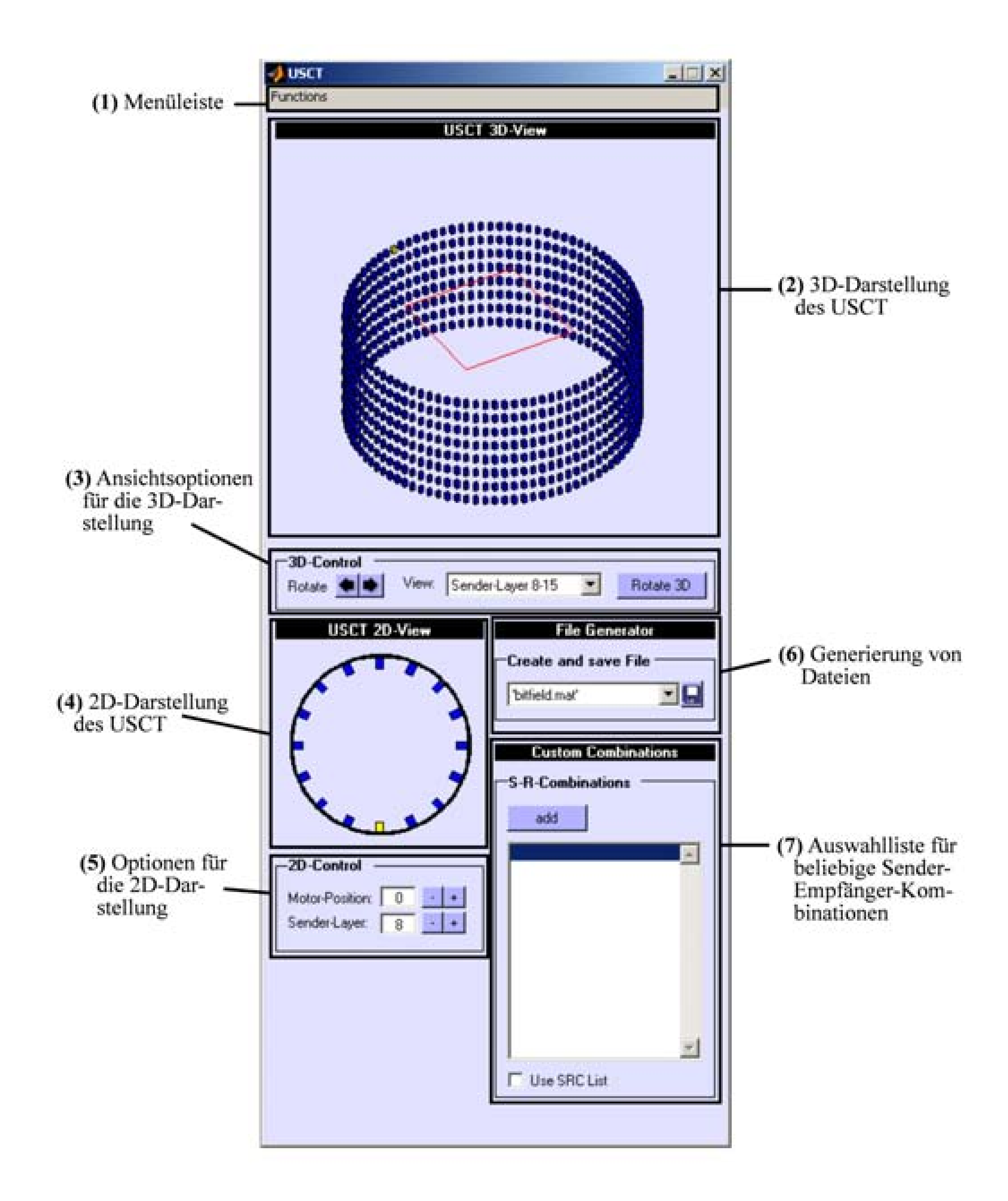


Abbildung 6.9: USCT-Oberfläche

2D-Modell an die entsprechende Stelle gedreht. Ebenfalls über zwei Schaltflächen und eine Editbox kann der dargestellte Senderlayer verändert werden.

Mit dem „File Generator“ (6) können Parameterdateien für andere Programme erzeugt werden. Die Art der zu erstellenden Datei kann über das Drop-Down-Menü ausgewählt werden. Die Generierung und Abspeicherung erfolgt durch Betätigen der Speichern-Schaltfläche, die mit einem Diskettensymbol versehen ist.

Bei mehreren ausgewählten Sendern und Empfängern im 3D-Modell bzw. den A-Scan-Auswahllisten werden automatisch alle möglichen Sender-Empfänger-Kombinationen generiert. Möchte der Benutzer jedoch mehreren Sendern jeweils andere Empfänger zuordnen, wird in der USCT-Oberfläche ein zusätzliches Bedienelement in Form eines Listenfeldes benötigt (7). Eine markierte Kombination im USCT-Modell oder den A-Scan-Listenfeldern kann über den „add“-Button in die Liste übernommen werden. Mit den im Kontextmenü verfügbaren Befehlen können einzelne, mehrere oder alle Elemente aus der Liste entfernt werden. Eine Checkbox dient dem Umschalten des Eingabeobjektes von den Auswahllisten in der A-Scan-Oberfläche auf die Liste in der USCT-Oberfläche. Das aktuelle Eingabeobjekt wird mit einem grünen Rahmen repräsentiert.

In der Menüleiste (1) ist der Unterpunkt „Functions“ zur Auflistung der verfügbaren Plugins der USCT-Oberfläche eingebunden.

6.2.2 Funktionalität

Wie auch die A-Scan-Oberfläche initialisiert die USCT-Oberfläche beim Starten zunächst einige Variablen, bindet die Pfade zu den Unterordnern ein und liest den Ordner der Menüleisten-Schnittstelle aus.

Die 3D-Visualisierung beruht auf der Geometrie-Datei, die die Position jedes Sendeelements im USCT als kartesische Koordinate enthält. Dieses wird innerhalb zweier vorgegebener Grenzen (Start- und Endlayer) durchlaufen und an jeder ausgelesenen Position ein Voxel gezeichnet, das über die gleichnamige Funktion[3] von *Joel Suresh* erzeugt werden kann. Die Voxel-Funktion erzeugt aus mehreren so genannten Patches (zweidimensionale Objekte, die über MATLAB-eigene Funktionen erzeugt werden können) ein dreidimensionales Objekt mit gewünschter Ausdehnung. Die Darstellung der erzeugten Voxel erfolgt in einem Axes-Feld.

Jedes erzeugte Voxel liefert eine Referenz auf sich zurück, welche in einer Matrix entsprechend dem Senderlayer und der Sendernummer gespeichert wird. Den Referenzen, die in MATLAB aus Handles-Objekten bestehen, können zusätzliche Daten angehängt werden. Hierfür dient das MATLAB-Konstrukt

```
setappdata(referenceToObject,variableNameAsString,value);
```

So wird jedem erzeugten Voxel zugewiesen, welches Sende- und welche Empfangselemente es repräsentiert. Der Senderlayer SL und die Sendenummer SN kann aus den Zählvariabeln der bei der Erzeugung durchlaufenen for-Schleifen entnommen werden.

Die Koordinaten der Empfängerelemente (RL sowie RN), die das Sendelement umgeben, lassen sich nach Formel 6.10 berechnen:

[3]Voxel-Funktion zur Erzeugung von dreidimensionalen Körpern, Quelle: `http://www.mathworks.com/matlabcentral/fileexchange`, Stand: 31.08.2006

$$RCoords = \begin{bmatrix} RL_{ul} = (SL \cdot 2) - 1 & RL_{ur} = (SL \cdot 2) - 1 \\ RN_{ul} = (SN \cdot 2) - 1 & RN_{ur} = SN \cdot 2 \\ & \\ RL_{ll} = SL \cdot 2 & RL_{lr} = SL \cdot 2 \\ RN_{ll} = (SN \cdot 2) - 1 & RN_{lr} = SN \cdot 2 \end{bmatrix} \quad (6.10)$$

wobei die Empfänger in der frontalen Sicht auf das TAS identisch zur abgebildeten Matrix angeordnet sind. Die Indizes sind als $_{ul}$ = Upper Left, $_{ll}$ = Lower Left, $_{ur}$ = Upper Right und $_{lr}$ = Lower Right zu interpretieren.

Gespeichert werden die Koordinaten in jeweils zweielementigen Arrays für die beiden Empfängerlayer und die beiden Empfängernummern.

Darüber hinaus wird dem Voxelobjekt die berechnete lineare Sender- und Receivernummer angehängt. Erstere berechnet sich nach Formel 6.11, Letztere nach Formel 6.12.

$$linearSender = SN + (SL \cdot 96) - 96 \quad (6.11)$$

$$linearReceiver = RN + (RL \cdot 192) - 192 \quad (6.12)$$

Jedem Voxel wird ein zuvor erzeugtes Kontextmenü (Abbildung 6.10) zugewiesen. Darin werden die folgenden Auswahlarten ermöglicht:

- Als Sender-Element markieren
- Als Receiver-Element markieren
- Ganzen Layer als Sender / Receiver markieren
- Ganze Spalte als Sender / Receiver markieren
- Deselektieren eines Elements, aller Elemente, eines Layers oder einer Spalte.

Die mit den Kontextmenü-Einträgen verbundenen Funktionen arbeiten nach einem nahezu identischen Prinzip. Zunächst werden anhand der erhaltenen Referenz des angeklickten Voxels die damit gespeicherten Daten wie Senderlayer, Sendernummer usw. ausgelesen. Anschließend kann über die Referenz und den set-Befehl die Farbe des Voxels geändert werden. Schließlich aktualisiert die Funktion die zentrale Datenhaltung.

Die zuletzt genannten Funktionalitäten (Einfärben des Voxels und Aktualisieren der Datenhaltung) sind in eine eigenständige Funktion ausgelagert, so dass ein Aufruf nach der Auswahl einer Sender-Empfänger-Kombination in der A-Scan-Oberfläche möglich ist, wodurch die 3D-Darstellung aktualisiert wird. Umgekehrt aktualisiert die durch das Voxel-Kontextmenü geladene Funktion die Markierungen in den Listenfeldern der A-Scan-Oberfläche. Damit ist eine konsistente Sicht auf die beiden Bedienelemente geschaffen.

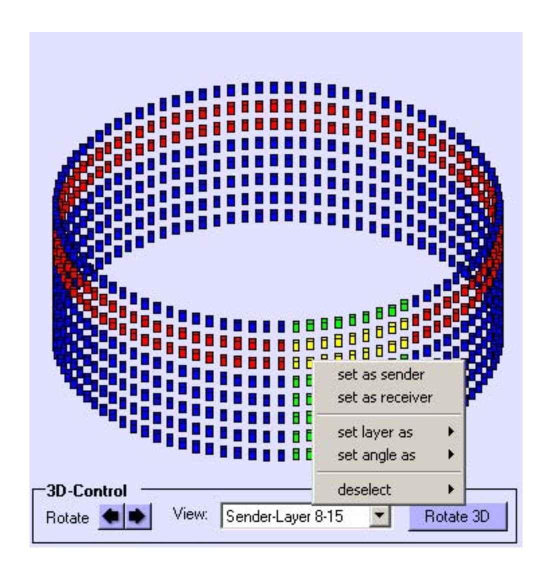


Abbildung 6.10: 3D-Modell des USCT mit markierten Elementen als Sender (rot), Empfänger (grün), Sender und Empfänger (gelb) sowie das auf die Elemente definierte Kontextmenü

Beim realen Ultraschall-Computertomographen ist - wie bereits erwähnt - ein Sende-Element von 4 Empfangs-Elementen umgeben. Da der Übersicht wegen die 3D-Visualisierung lediglich die Sende-Elemente als Voxel darstellt, ist es erforderlich, zusätzlich eine Möglichkeit zur Auswahl eines einzelnen Empfangselementes zu ermöglichen. Dies geschieht über einen kleinen Dialog, der bei der Auswahl „set as Receiver“ geöffnet wird. In ihm werden die vier dem Sendeelement zugeordneten Empfangselemente (Berechnung nach Formel 6.10) als klickbare Patches dargestellt (Abbildung 6.11). Ein aktivierter Empfänger wird durch eine grüne Färbung des Patches visualisiert, deaktivierte Empfangselemente haben eine blaue Farbe. Die Funktion setzt die globalen Variablen zur Speicherung der markierten Empfänger.

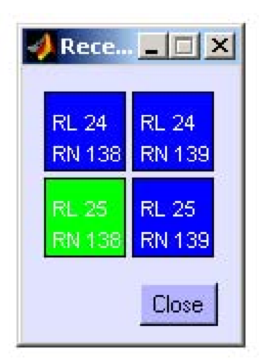


Abbildung 6.11: Dialog zur Auswahl von Receiverelementen

Die Ansicht des 3D-Modells kann über zwei Schaltflächen gedreht werden. Die jeweiligen Callback-Funktionen ändern dabei die Ausrichtung des gesamten Axes-Objektes, wodurch ein zeitaufwendiges Neuzeichnen der Voxel vermieden wird. Mit dem Togglebutton „Rotate“ wird für das Axes-Objekt der Rotationsmodus ein- bzw. ausgeschaltet, der ein beliebiges Drehen desselben in alle Richtungen ermöglicht.

Aufgrund der großen Anzahl von Sendeelementen und der daraus resultierenden Fülle von Voxeln, kann die Übersichtlichkeit des Modells schnell verloren gehen. Um dem entgegenzuwirken ist das 3D-Modell in drei Schichten entsprechend der TAS-Layer unterteilt. Diese sind im Drop-Down-Menü auswählbar. Beim Wechseln des Layers wird die Zeichen-Funktion des 3D-Modells wie oben beschrieben mit veränderten Parametern für Beginn- und End-Senderlayer aufgerufen.

Eine Vereinfachung der Auswahl von Sendern und Empfängern wird durch eine programmierbare linke Maustaste erreicht. Den Voxeln werden neben den Kontextmenüs so genannte „ButtonDownFunctions“ zugewiesen, die aufgerufen werden, sobald mit einer beliebigen Maustaste auf sie geklickt wird. In der zugehörigen Funktion kann dann überprüft werden, welche Maustaste zuletzt gedrückt wurde.

Beim Aufruf einer Funktion über das Kontextmenü der Voxel wird der Funktionsname in einer Variablen zwischengespeichert. Ein Klick mit der linken Maustaste bewirkt nun eine Abfrage dieser Variablen mit anschließender Ausführung der Funktion. Dabei ändert sich durch den Mausklick die Referenz des aufrufenden Voxels, so dass die zuletzt ausgeführte Operation auf das angeklickte Voxel angewendet wird.

Dem Benutzer erspart diese Funktion das recht aufwendige Navigieren durch das Kontextmenü, welches wiederum notwendig ist, um dem Benutzer verschiedene Auswahlmuster zur Verfügung zu stellen. Damit ist es möglich ganze Spalten (24 Sender- bzw. 48 Empfängerelemente) oder ganze Ebenen (96 Sende- bzw. 192 Empfängerelemente) mit nur einem einzigen Mausklick zu markieren. Dies führt zu einer sehr schnellen Auswahl einer großen Anzahl von Elementen.

Während das 3D-Modell vor allem der guten Visualisierung und schnellen Selektion von Sender-Empfänger-Elementen dient, kann mit dem 2D-Modell präziser auf bestimmte Sender- und Empfänger-Nummern zugegriffen werden. Das 2D-Modell (Abbildung 6.12) stützt sich wie auch das 3D-Modell auf Patches, die jedoch nicht zu Voxeln zusammengesetzt, sondern als einzelne Rechtecke verwendet werden. Jedes Rechteck repräsentiert dabei konsistent zur 3D-Ansicht einen Sender sowie vier Empfänger.

Die Ansicht wird gezeichnet, indem jeweils ein Patch an einer Stelle in das Axes-Objekt gezeichnet und anschließend um den Winkel zwischen zwei realen Sender-Elementen (22,5°) rotiert wird. Anschließend wird wieder ein Patch an der gleichen Stelle gezeichnet usw. Eine aufwendige Berechnung der Positionen

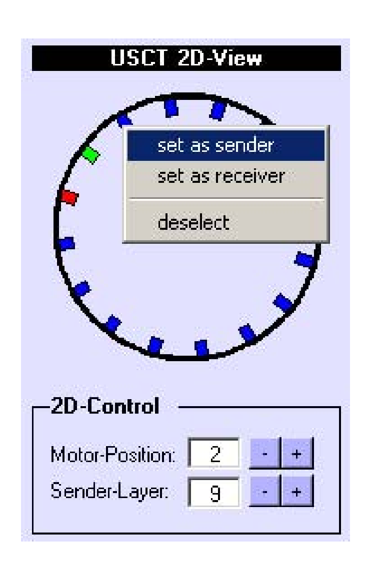


Abbildung 6.12: 2D-Modell des Ultraschall-Computertomographen

kann somit entfallen.

Jedem Patch wird erneut eine Nummer zugeordnet, über die er identifiziert und referenziert werden kann. Wie auch bei der 3D-Ansicht wird für die Elemente ein Kontext-Menü definiert, das die hinterlegte Information ausliest und damit weitere Aktionen ausführen kann. Diese Funktionen werden im Kontext-Menü zur Auswahl gestellt und umfassen:

- Ein Element als Sender markieren
- Ein Element als Empfänger markieren
- Deselektieren eines Elements

Entsprechend der Auswahl werden die Patches eingefärbt. Bei der Auswahl „set as Receiver“ wird der vom 3D-Modell bereits bekannte Dialog zur Auswahl eines Empfängerelements geöffnet.

Analog zu den Update-Funktionen des 3D-Modells gibt es Update-Funktionen für das 2D-Modell, die ein Mapping zwischen den Nummern der markierten Sender und Empfänger vornehmen und die Farben sowie die Datenhaltung bei einer Auswahl in den A-Scan-Auswahllisten oder dem 3D-Modell aktualisieren.

Ein markierter Sender bzw. Empfänger in den Auswahllisten oder dem 3D-Modell liegt dabei in virtueller Nummerierung ohne Berücksichtigung der Motorpositionen vor. Das Mapping auf die im 2D-Modell abgebildeten realen Sendeelemente erfolgt nach Formel 6.13 für Sender und 6.14 für Empfänger.

$$SE_{real} = \left\lfloor \frac{SE_{virtual}}{6} \right\rfloor \tag{6.13}$$

$$SE_{real} = \left\lfloor \frac{\left\lfloor \frac{RE_{virtual}}{2} \right\rfloor}{6} \right\rfloor \tag{6.14}$$

Entgegengesetzt werden aus Markierungen im 2D-Modell die Auswahllisten der A-Scan-Oberfläche sowie die 3D-Ansicht aktualisiert.

Die Drehung der 2D-Ansicht über die Änderung der Motorposition wird durch eine MATLAB-Funktion zur Rotation der Ansicht durchgeführt. Dabei werden die Winkel von 3,75° wie beim realen USCT verwendet. Der Wechsel des Senderlayers erfolgt durch das Setzen einer globalen Variable anhand des im zugehörigen Feld ermittelten Wertes.

Für andere Softwaresysteme der USCT-Abteilung wie z.B. die Datenakquisitionssoftware oder die Rekonstruktionssoftware werden Parameterdateien in verschiedenen Formaten benötigt, die die markierten Sender-/Empfänger-Nummern beinhalten:

- AvailableEmitters.txt
- AvailableReceivers.txt
- Bitfield.mat

Entprechend des in Abschnitt 4.1.3 beschriebenen Formates wird eine Funktion implementiert, die die markierten Sender und Empfänger aus der globalen Variablen ausliest, nach Formel 6.11 bzw. 6.12 in lineare Nummerierung umrechnet und schließlich in der beschriebenen Formatierung in eine Textdatei schreibt. Zuvor wird ein Speichern-unter-Dialog aufgerufen, der dem Benutzer ermöglicht einen Dateinamen (Standard: AvailableEmitters bzw. AvailableReceivers) und einen Pfad anzugeben.

Der Aufbau des Bitfields gestaltet sich etwas aufwendiger, da hier ein Zugriff auf Bitebene notwendig ist. Nach dem Auslesen der globalen Matrizen mit den markierten Sendern und Empfängern muss zunächst die Stelle n im Bitfield berechnet werden, an der das Bit gesetzt werden muss. Dies folgt dem umgekehrten Weg wie in den Formeln 4.2-4.5 und berechnet sich aus dem daraus abgeleiteten Ausdruck 6.15.

$$n = (SL \cdot 192 \cdot 48 \cdot 96) + (SN \cdot 192 \cdot 48) + (RL \cdot 192) + RN + 1 \tag{6.15}$$

Zur Speicherung des Bitfields wird ein Array von uint8-Variablen verwendet, d.h. pro Arrayelement können acht Bitfieldstellen gespeichert werden. In welchem Arrayelement die Stelle n liegt kann durch Formel 6.16 ermittelt werden:

$$arrayIndex = \left\lfloor \frac{(n-1)}{8} \right\rfloor + 1 \tag{6.16}$$

Mit dem MATLAB-Befehl „bitset“ kann nun im errechneten Arrayelement an der Stelle *bitIndex* eine 1 oder 0 gesetzt werden. Die Stelle *bitIndex* wird über Formel 6.17 bestimmt:

$$bitIndex = \begin{cases} 9 - (n \bmod 8) & \text{für } n \bmod 8 \neq 0 \\ 1 & \text{für } n \bmod 8 = 0 \end{cases} \tag{6.17}$$

In das Listenfeld „Custom Combinations“ kann durch Klicken auf „add“ die aktuelle Auswahl von Sendern und Empfängern hinzugefügt werden. Im Hintergrund speichert das Listenfeld dabei die Nummern der Senderlayer, Sendernummern, Empfängerlayer und Empfängernummern in der UserData, einer speziellen Variable, die dem Bedienfeld zugeordnet ist, in tabellarischer Form wie bereits in Tabelle 6.1 beschrieben. Im Vordergrund wird dem Benutzer im Listenfeld ein String angezeigt, der aus den genannten Nummern generiert wird (Abbildung 6.13).

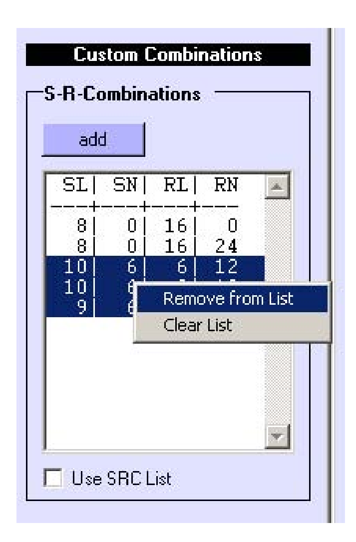


Abbildung 6.13: Listenfeld zur Auswahl und Anzeige beliebiger Sender-Empfänger-Kombinationen

Die Liste kann sowohl als Eingabeobjekt für das Laden von A-Scans und die damit verbundenen weiteren Aktionen verwendet werden, wie auch als Ausgabefenster für Berechnungen, die eine Liste von A-Scans als Ergebnis haben.

Mit den Funktionen im Kontextmenü können Einträge aus der Liste gelöscht werden. Die Callback-Funktion zum Befehl „Remove from List“ durchläuft dabei die gesamte Liste und löscht markierte Zeilen, bevor der neue String in das Listenfeld geschrieben und die userData-Variable aktualisiert wird. „Clear List“ setzt den Inhalt des Listenfeldes sowie die Variable userData zurück.

Mit der Option „Use SRC List“ kann analog zur „Use TAS List“-Option in der A-Scan-Oberfläche ausgewählt werden, welche Liste als Eingabeobjekt zum Laden von A-Scans dient. Dabei wird ein Flag gesetzt, das in der Lade-Funktion der A-Scan-Oberfläche überprüft wird.

6.3 Rekonstruktionsbilder-Oberfläche

Als dritter Hauptteil der Software wird die Oberfläche für rekonstruierte Bilder (Abbildung 6.14) implementiert. Dabei wird ein Konzept verfolgt, das zwei Bild-"Kanäle" bietet, d.h. es können gleichzeitig zwei Bilder in die Oberfläche geladen werden. Die Anzeigeobjekte und darauf operierende Funktionen werden mit einer Art Schalter auf das jeweilige Bild umgeschaltet.

6.3.1 GUI

Die Menüleiste (1) beinhaltet erneut ein Menü zur Auflistung der über die Schnittstelle eingebundenen Plugins.

Zum Anzeigen der Bilder wird ein Axes-Objekt verwendet, das sich nahezu über die gesamte Breite der Oberfläche erstreckt (2). Die Konfiguration ist auf die Anzeige von Bildern ausgelegt, so dass die Seitenverhältnisse eines Bildes erhalten bleiben.

Bei der Erstellung von Schichtbildern muss die jeweilige Schnittebene und die Höhe selbiger im Volumen angegeben werden. In der Oberfläche erfolgt die Auswahl der Orientierung aus den drei Hauptachsenkombinationen über drei sich gegenseitig verriegelnde Toggle-Buttons (4). Die Schicht wird durch eine Bildlaufleiste (3) rechts des Anzeigefeldes ausgewählt.

Durch den Colorbar-Togglebutton (5) kann die MATLAB-Colorbar in das Anzeigefeld ein- und ausgeblendet werden. Sie visualisiert die Skalierung der Farben, die im angezeigten Bild verwendet werden, so dass der Benutzer die aufsummierten Amplitudenwerte innerhalb des rekonstruierten Bildes ablesen kann.

Das Selektionswerkzeug (6) ermöglicht die Markierung von Bildausschnitten, die dann für weitere Aktionen verwendet werden können. Nach Drücken der Schaltfläche und Klicken in das angezeigte Bild wird ein Rahmen gezeichnet, der auf eine beliebige Größe gezogen werden kann. Die Markierung bleibt auch beim Wechsel der Schichten erhalten. Eine Checkbox im gleichen Rahmen wählt aus, ob nach der Markierung einer Region im Bild die Zurückrechnung auf einen Bereich im A-Scan (siehe hierfür 6.4.2) ausgeführt werden soll.

Zusammengefasst in einem Rahmen (7) finden sich alle Funktionen, die sich mit dem Laden, Anzeigen und Abspeichern der rekonstruierten Bilder beschäftigen. Zwei identische Steuereinheiten repräsentieren die beiden Bildkanäle. Über die „Load Image"-Schaltflächen kann ein Bild in die Oberfläche geladen werden. Die im Datensatz vorhandenen Bildarten können über jeweils vier Radiobuttons ausgewählt werden. Dabei sind die Namen der Bildarten über die benutzerdefinierten Einstellungen änderbar. Die Schaltflächen mit der Bezeichnung „1" bzw. „2" ermöglichen das Wechseln auf den jeweiligen Bildkanal. Visualisiert wird die Auswahl durch eine grüne Einfärbung der Schaltflächen. „Export Image" erlaubt es, das derzeit angezeigte Bild als Grafik im JPEG-Format oder als MATLAB-Datei abzuspeichern. Verbindungslinien zwischen den Schaltflächen visualisieren

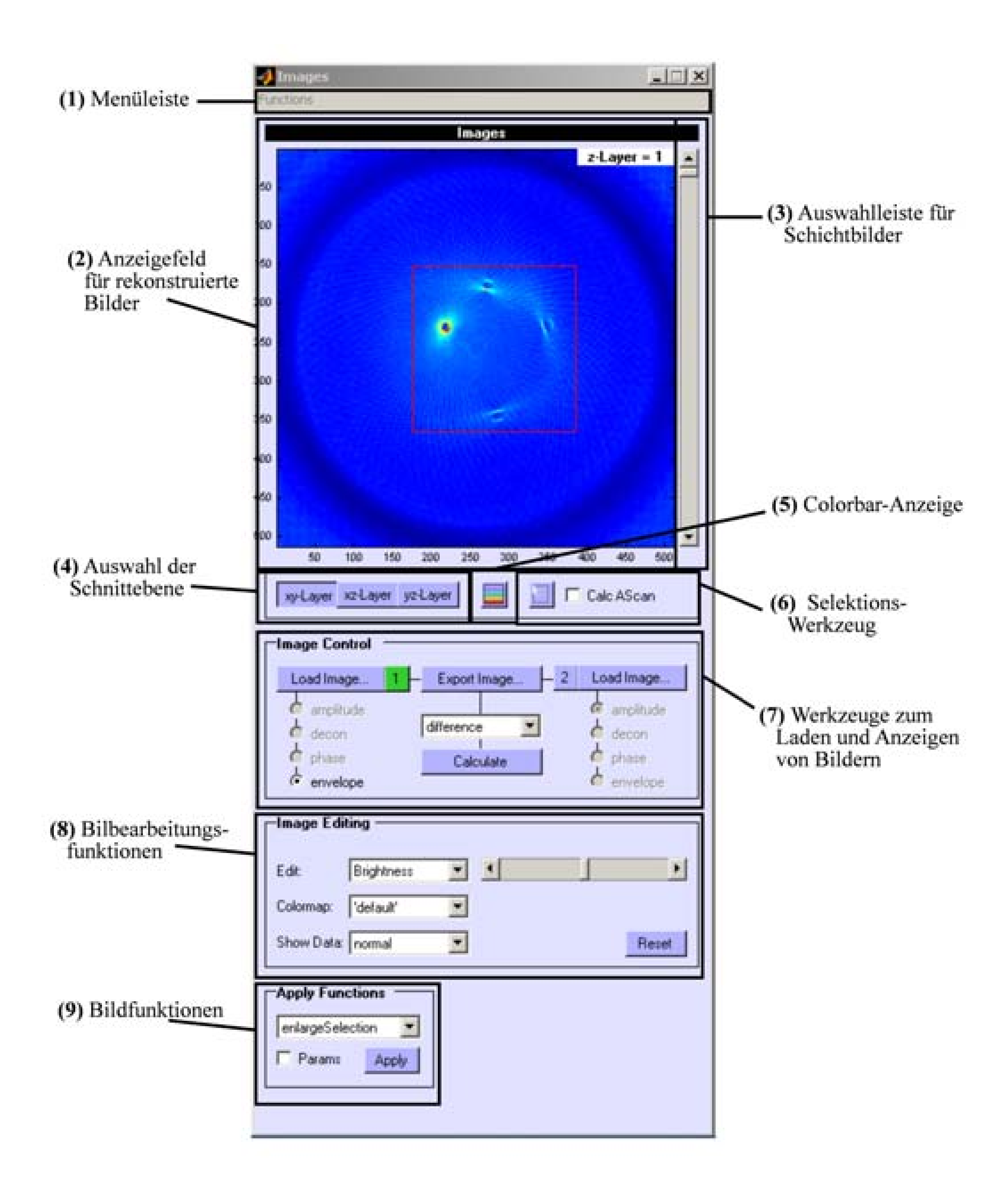


Abbildung 6.14: Oberfläche für rekonstruierte Bilder

die Funktionalität des Arbeitens auf zwei unterschiedlichen Bildern mit der gleichen Operation. Das Drop-Down-Menü dient der Auswahl einer Bildvereinigungsfunktion, welche durch das Klicken auf die Schaltfläche „Calculate“ ausgeführt wird. Das berechnete Bild wird dann im Anzeige-Fenster dargestellt, die Schaltfläche grün eingefärbt. Bildinformationen sind über die Tooltips der beiden Buttons „1“ und „2“ erhältlich.

Die Bildbearbeitungsfunktionen sind ebenfalls in einem Rahmen (8) zusammengefasst. Im obersten Drop-Down-Menü kann die Art der Bildbearbeitung ausgewählt werden. Mit der rechts daneben angeordneten Bildlaufleiste wird die Bildbearbeitung schließlich ausgeführt. Damit kann eine stufenlose Anpassung der Bildeigenschaften wie z.B. der Helligkeit erfolgen. Zwei weitere eigenständige Drop-Down-Menüs bieten dem Benutzer die Möglichkeit, die Colormap, mit der das Bild dargestellt wird, zu verändern, sowie die Daten logarithmiert anzuzeigen. Eine Schaltfläche „Reset“ versetzt das geladene Bild in den Ausgangszustand.

Die Schnittstelle für beliebige Funktionen auf den Bildern bzw. Bildausschnitten (9) benutzt ein Drop-Down-Menü zur Auswahl der auszuführenden Operation. Über eine Checkbox kann gewählt werden, ob eventuell vorhandene zusätzliche Parameter vor dem Aufruf der Funktion verändert werden sollen. Mit der Schaltfläche „Apply“ wird die Funktion schließlich aufgerufen.

6.3.2 Funktionalität

Für das Konzept der beiden Bildkanäle werden zentrale Variablen für die Bilddaten sowie die Informationen über das Bild eingerichtet, in denen entsprechend der Auswahl die Daten des aktuellen Bildes stehen. Beim Wechseln zwischen den Bildern werden die Variablen neu gesetzt.

In den Callback-Funktionen der „Load Image“-Schaltflächen wird zunächst ein Öffnen-Dialog angezeigt, mit dem der Benutzer eine Datei mit den Daten eines rekonstruierten Bildes auswählen kann. Anschließend wird die Funktion „loadImage“ aufgerufen, die die Datei einliest, nach den vorgegebenen (und in den benutzerdefinierten Einstellungen änderbaren) Variablennamen für die verschiedenen Bildarten sucht und die Daten zurückliefert. Zusätzliche Rückgabewerte sind Flags, welche die vorhandenen Bildarten beschreiben.

Darauffolgend versucht die Software die Info-Datei zum jeweiligen Bild zu laden. Alle Daten (Bilder und Infos) werden für jeden Bildkanal zwischengespeichert. Ein Volumenbild liegt dabei beispielsweise in Form eines dreidimensionalen Arrays vor.

Entsprechend der vorhandenen Bildarten werden die Radiobuttons zur Auswahl dieser freigeschaltet. Bei einem erfolgreichen Laden schaltet die Funktion außerdem die Bildbearbeitungs-Elemente frei. Anhand einer Funktion zur Erkennung der Dimension der Bilddaten werden auch die Toggle-Buttons zur Auswahl der Schnittebene freigeschaltet (bei 3D-Bildern) bzw. im deaktivierten Zustand belassen (2DBilder).

Einige Informationen über das geladene Bild wie z.B. die Größe, den Aufnahmemodus und die physikalische Auflösung werden in einen String zusammengesetzt und der Schaltfläche „1“ bzw. „2“ als Tooltip gesetzt.

Initial wird nach dem Laden eines Bildes die zentrale Variable für das aktuelle zur Verarbeitung und Anzeige vorgesehene Bild gesetzt und anschließend

je nach Dimension des Bildes ein Schnittbild anhand der mit der Bildlaufleiste ausgewählten Ebene erzeugt und angezeigt. Die Bildlaufleiste wird zuvor auf den Wertebereich angepasst, der die Anzahl der im Volumenbild enthaltenen Ebenen umfasst. Das Schnittbild berechnet sich durch das „Herausschneiden“ der Daten zweier Dimensionen an der ausgewählten Stelle in der dritten Dimension des Datenarrays. Mit dem MATLAB-Befehl „squeeze“ wird zuletzt die nun leere dritte Dimension entfernt.

Das Schnittbild wird durch die Subfunktion „showSingleImage“ dargestellt. Die Bilddaten liegen transponiert vor, da es sich beim USCT-Koordinatensystem im Gegensatz zur MATLAB-Darstellung um ein rechtshändiges System handelt. Dies wird durch erneutes Transponieren der Bildmatrix aufgehoben. Bei Bildern aus der XZ- bzw. YZ-Ebene entfällt dieser Schritt. Die Darstellung der Daten erfolgt als indexiertes Bild, wobei den Zahlenwerten der Matrix Farbwerte der aktuellen Colormap zugeordnet werden.

Beim Wechsel zwischen den beiden Bildkanälen werden die zentralen Bild-Variablen sowie die Variablen zur Speicherung der Bildinformationen mit den jeweiligen Daten des Bildkanals überschrieben. Anschließend wird erneut die Routine zur Berechnung und Anzeige von Schichtbildern aufgerufen. Darüber hinaus werden die Bedienelemente der Oberfläche aktualisiert, wenn beispielsweise durch einen Wechsel auf ein 2D-Bild die Bildlaufleiste nicht mehr benötigt wird.

Der Wechsel zwischen den Bildarten wird in den Callback-Funktionen der Radiobuttons geregelt. Dabei wird eine Variable mit der Bezeichnung der Bildart gefüllt und die zentrale Variable des angezeigten Bildes mit den jeweiligen Daten gefüllt. Der Aufruf der Anzeigeroutine stellt das Bild dar.

Der Wertebereich der Bildlaufleiste ist abhängig von der Größe eines Bildes und der derzeitigen Schnittrichtung. Eine Subfunktion der Oberfläche setzt den Wertebereich, sobald zwischen Bildern gewechselt wird, ein Bild geladen oder die Schnittrichtung geändert wird. Der beim Bewegen der Bildlaufleiste zurückgegebene Wert wird in einer Variablen zwischengespeichert und die Funktion zur Erzeugung eines Schnittbildes mit ihm aufgerufen. Somit hat der Benutzer die Möglichkeit ein Volumen gezielt zu durchlaufen. Der Wert wird zusätzlich in einem überlagerten Textfeld in der rechten oberen Ecke des Bildes ausgegeben.

Für Volumenbilder kann die Sichtweise, d.h. die Schnittrichtung (Abbildung 6.15) gewechselt werden. Dabei wird eine Variable mit einem die Schnittrichtung repräsentierenden Zahlenwert aktualisiert, der in der Routine zur Berechnung von Schichtbildern berücksichtigt wird.

In der Callback-Funktion der Export-Schaltfläche wird der Benutzer aufgefordert in einem Speichern-unter-Dialog einen Dateinamen anzugeben. Anhand dessen Endung wird dann innerhalb der Funktion entschieden, ob die Bilddaten als MATLAB-Datei oder als JPEG-Bild exportiert werden.

Ersterer Fall benutzt die 2D-Daten des aktuellen Schichtbildes und kopiert sie in eine Variable, die den Namen der derzeit ausgewählten Bildart trägt. Darauf

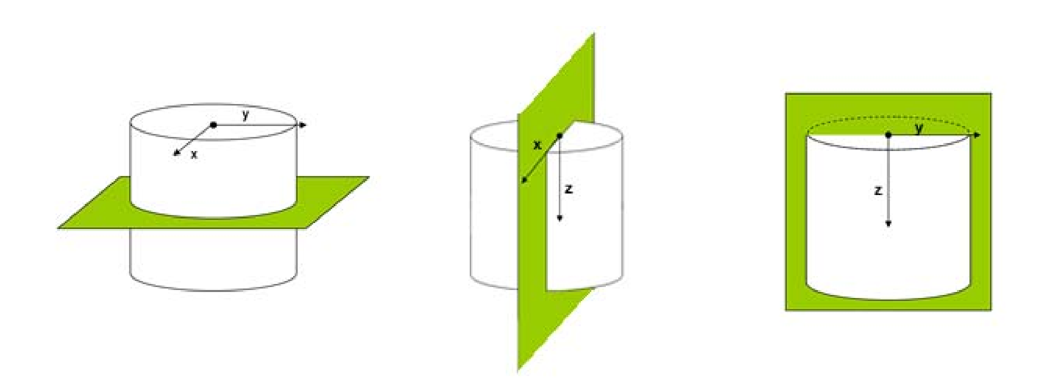


Abbildung 6.15: Schnittrichtungen für die Erzeugung von Schichtbildern aus einem Volumen

folgend wird die Variable im MATLAB-Format auf der Festplatte abgespeichert. Der Export in eine JPEG-Datei erfolgt über die MATLAB-Funktion „imwrite“, die eine Variable als Eingabewert verlangt, in der die Farbwerte im RGB-Format stehen. Deshalb ist zunächst eine Umrechnung der Matrixwerte in mehreren Schritten notwendig:

- Auslesen der aktuellen Colormap, die für das indexierte Bild verwendet wird.
- Die Größe dieser Colormap gibt an, auf wie viele Farbwerte das Datenarray des geladenen Bildes abgebildet wird.
- Mapping der Bilddaten b mit den Bildpunkten $x_{i,j}$ auf die Größe der Colormap:

$$b = \begin{bmatrix} x_{1,1} & \cdots & x_{1,n} \\ \vdots & \ddots & \vdots \\ x_{m,1} & \cdots & x_{m,n} \end{bmatrix} \tag{6.18}$$

$$\forall x \in b : x = \frac{x}{min(b)} - 1 \Rightarrow b' \tag{6.19}$$

Der geringste Wert in b' nimmt nun den Wert 0 an, alle anderen Werte verteilen sich entsprechend skaliert.

$$\forall x \in b' : x = \frac{x}{max(b')} \Rightarrow b'' \tag{6.20}$$

Der größte Wert in b'' nimmt nun den Wert 1 an, alle anderen Werte verteilen sich entsprechend. Die Daten des rekonstruierten Bildes wurden also auf

das Intervall $[0, 1]$ abgebildet. Diese müssen nun auf die Größe der Colormap skaliert werden:

$$\forall x \in b'' : x = x \cdot size(colormap) \Rightarrow b''' \tag{6.21}$$

- Die Daten aus b''' können nun der MATLAB-Funktion ind2rgb übergeben werden, die daraus ein RGB-Bild erzeugt, welches mit oben beschriebener Funktion abgespeichert werden kann.

Die Schnittstelle für Bildvereinigungsfunktionen basiert auf dem mittlerweile bekannten Konzept des Auslesens eines Ordners, in dem die Funktionen abgelegt sind. Dies geschieht beim Start der Oberfläche, gefolgt vom Schreiben der Funktionsnamen in das Drop-Down-Menü. Eine Auswahl in diesem setzt eine Variable mit dem Namen der Funktion. Über den Button „Calculate“ wird eben diese Funktion aufgerufen.

Bildvereinigungsfunktionen erhalten als Parameter die Volumen- bzw. Schichtbilddaten der beiden geladenen Bilder. Als Rückgabewert ist es für den Programmierer Vorschrift, ein zwei- bzw. dreidimensionales Datenarray zu liefern, das anschließend wie ein geladenes Bild in der Oberfläche behandelt werden kann. Weitere Vorgaben werden bei dieser Schnittstelle nicht gemacht. Der Programmierer hat hier freie Hand in der Implementierung, muss also beispielsweise auch Fehlerbehandlungen selbst vornehmen.

Bei den Bildbearbeitungen wird eine Funktion zum Einstellen der Helligkeit des Bildes angeboten. Dafür wird die MATLAB-Funktion „brighten“ verwendet, die lediglich die Colormap des indexierten Bildes anpasst und die Daten unverändert lässt. Mit der Bildlaufleiste rechts neben dem Drop-Down-Menü kann der Wert, der der Funktion übergeben wird, ausgewählt werden. Diese erwartet Werte zwischen 0 und 1 für die Aufhellung des Bildes bzw. Werte zwischen -1 und 0 für die Abdunklung des Bildes. Da die Bildlaufleiste bei jeder Bewegung einen Wert zwischen 0 und 1 liefert, muss in seiner Callback-Funktion der Wert auf den Unterschied zur vorherigen Auswahl reduziert werden, um somit absolute Werte erreichen zu können.

Eine weitere Bildbearbeitungsfunktion ist die Anpassung der Colormap. Hier bietet MATLAB einige vordefinierte Varianten an, welche im Drop-Down-Menü aufgelistet werden. Durch die Auswahl einer Zeile darin wird die derzeit verwendete Colormap durch die Neue ersetzt. Das angezeigte Bild verändert dabei sofort das Aussehen. Zusätzlich wird die Bildlaufleiste für die Helligkeit zurückgesetzt, da die Colormap neu initialisiert wurde und Anpassungen der Helligkeit an der vorherigen Colormap verloren gehen.

Als dritte Bildbearbeitungsfunktion hat der Benutzer die Möglichkeit, Daten eines rekonstruierten Bildes in logarithmischer Verteilung anzuzeigen. Dazu werden die Bilddaten in der Funktion zur Berechnung eines Schichtbildes, abhängig

von dieser Auswahl logarithmiert oder nicht. Die Auswahl im Drop-Down-Menü setzt dabei ein Flag, das beim Generieren von Schichtbildern berücksichtigt wird.

Für die Markierung einer „Region of Interest“ im angezeigten Bild wird die Funktion „rubberbandbox“[4] von *Bob Hamans* verwendet. Diese liefert die Koordinaten des linken oberen und linken unteren Eckpunktes einer Selektion, welche in einer globalen Variablen nach dem Format aus Formel 6.22 gespeichert werden.

$$selection = \begin{bmatrix} x_{ul} & y_{ul} \\ x_{lr} & y_{lr} \end{bmatrix} \tag{6.22}$$

In der Callback-Funktion der Colorbar-Schaltfläche wird die MATLAB-Funktion „colorbar“ mit den Argumenten „on“ bzw. „off“ verwendet, um die beim indexierten Bild verwendete Colorbar ein- bzw. auszublenden.

Die Schnittstelle zur Anwendung von Funktionen auf Bildausschnitten ähnelt vom Aufbau der Schnittstelle für A-Scan-Transformationen. Die aus dem Unterordner ausgelesenen Funktionen werden beim Start der Software in das Drop-Down-Menü geschrieben. Bei einer Auswahl in selbigem wird der Name der Funktion in eine Variable geschrieben, auf die die „Apply“-Schaltfläche zugreift und die Funktion entsprechenden Namens ausführt. Als ersten Parameter erhält diese zunächst die aktuellen Bilddaten, welche als Standardeinstellung aus den 2D-Daten des dargestellten Layers bestehen. Über einen Eintrag in der Parameterdatei lässt sich diese Option auch auf die Übergabe des 3D-Datensatzes abändern. Zweiter Parameter ist die Nummer des Layers, der jedoch nur bei einem 3D-Datensatz relevant ist. Bei 2D-Daten wird der Wert 1 übergeben. Als dritter Parameter werden die Koordinaten der ROI übergeben.

Alternativ erhält die Funktion als vierten Parameter das Struct „inputParams“. Spezielle Bedeutung hat dabei der Parameter „inputParams.dataMode“. Mit diesem kann ausgewählt werden, ob eine Funktion lediglich die 2D-Daten des aktuell dargestellten Bildes erhält (Wert '2D'), oder ob sie den kompletten 3D-Datensatz übergeben bekommt (Wert '3D'). Eine der Schnittstelle entsprechende Funktion hat keinen Rückgabewert, so dass sich der Programmierer mit der Visualisierung der Ergebnisse und der Fehlerbehandlung beschäftigen muss. Zum Ändern des Parameters „inputParams“ wird, wie bei den A-Scan-Transformationen, die Funktion „readInParameters“ verwendet.

6.4 Oberflächenübergreifende Funktionen

6.4.1 Markierung der Bildselektion im 3D-USCT

Die Position einer Selektion im rekonstruierten Bild wird zur besseren Orientierung des Benutzers in das 3D-Modell eingezeichnet. Er kann nun erkennen aus

[4]Funktion „rubberbandbox“ zur Auswahl eines Ausschnittes in einem Axes-Objekt. Quelle: `http://www.mathworks.com/matlabcentral/fileexchange`, Stand: 02.09.2006

welchem Bereich des realen USCT die ausgewählte Region im Bild stammt und kann das Aussehen des Bildes mit seinem Wissen über die Charakteristik des USCT an dieser Stelle verknüpfen.

Für die Bestimmung der Position werden die Koordinaten der beiden Eckpunkte der in einer globalen Variablen gespeicherten ROI links oben und rechts unten anhand der Informationen über Auflösung und Startpunkt des Bildes auf das USCT-Koordinatensystem umgerechnet, nach welchem auch das 3D-Modell aufgebaut ist. Bei der Berechnung muss die Schnittrichtung des dargestellten Bildes berücksichtigt werden. So ergeben sich bei der Draufsicht in XY-Richtung die Formeln 6.23 bis 6.25

$$x_{USCT} = x_{Startp} + (x_{IMAGE} \cdot res) - \left(\frac{usctDiam}{2}\right) \tag{6.23}$$

$$y_{USCT} = -\left(y_{Startp} + (y_{IMAGE} \cdot res) - \left(\frac{usctDiam}{2}\right)\right) \tag{6.24}$$

$$z_{USCT} = z_{Startp} + ((layer - 1) \cdot res) \tag{6.25}$$

wobei x_{USCT}, y_{USCT} und z_{USCT} die Koordinaten im kartesischen Koordinatensystem des USCT darstellen, x_{IMAGE}, y_{IMAGE} und z_{IMAGE} die Koordinaten eines Eckpunktes der Selektion im Bild, sowie x_{Startp}, y_{Startp} und z_{Startp} die Koordinaten des Startpunktes der Rekonstruktion im kartesischen Koordinatensystem des USCT. Mit res ist die Auflösung des Bildes in $m/Pixel$ angegeben, $usctDiam$ bezeichnet den Durchmesser des USCT-Zylinders und $layer$ steht für die ausgewählte Schnittebene des Bildes.

Der Radius des USCT muss, aufgrund der Verschiebung des Nullpunktes in x- und y-Richtung bei der Rekonstruktion eines Bildes, abgezogen werden. Eine Invertierung der y-Koordinaten ist erforderlich, da durch die Spiegelung der Daten während der Darstellung eines Schnittbildes die Position im USCT ebenfalls gespiegelt berechnet werden würde. Bei den z-Werten ergibt sich aufgrund der nur zweidimensionalen Auswahl die Besonderheit, dass beide den gleichen Wert annehmen.

In XZ-Richtung ändert sich die Berechnung der y- und z-Koordinate aufgrund der Darstellung der Schnittbilder im MATLAB-Koordinatensystem wie folgt:

$$y_{USCT} = -\left(y_{Startp} + ((layer - 1) \cdot res) - \left(\frac{usctDiam}{2}\right)\right) \tag{6.26}$$

$$z_{USCT} = z_{Startp} + (y_{IMAGE} \cdot res) \tag{6.27}$$

In YZ-Richtung wird die z-Koordinate nach Formel 6.27 berechnet, während die Berechnung der x- und y-Koordinaten abgeändert werden muss:

$$x_{USCT} = x_{Startp} + ((layer - 1) \cdot res) - \left(\frac{usctDiam}{2}\right) \tag{6.28}$$

$$y_{USCT} = -\left(y_{Startp} + (x_{IMAGE} \cdot res) - \left(\frac{usctDiam}{2}\right)\right) \tag{6.29}$$

Aus den errechneten Koordinaten kann nun ein Rechteck in das 3D-Modell des USCT geplottet werden (Abbildung 6.16). Die beschriebene Umrechnung erfolgt automatisch beim Markieren einer Region im rekonstruierten Bild.

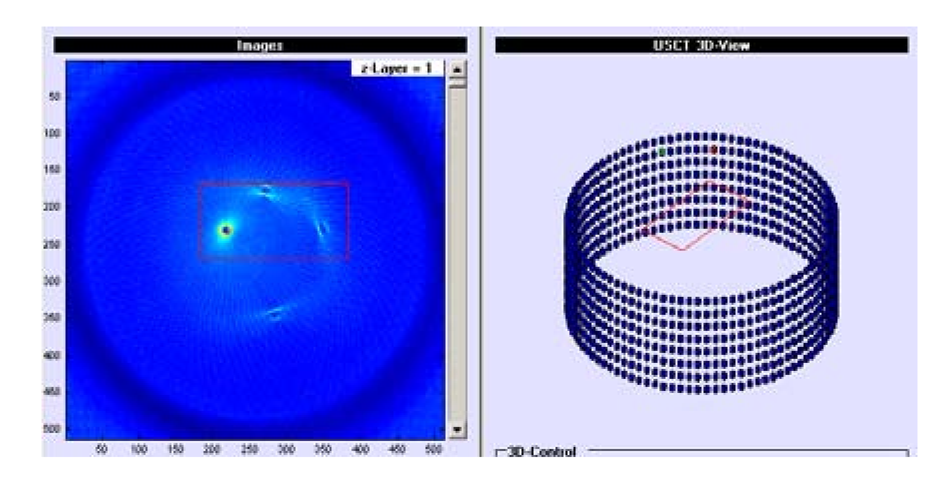


Abbildung 6.16: Die Auswahl eines Bereiches im rekonstruierten Bild (links) wird an der entsprechenden Stelle im 3D-Modell des USCT (rechts) angezeigt

6.4.2 Zurückrechnung einer Bildregion auf einen A-Scan-Bereich

Wie in Kapitel 4 beschrieben, basiert ein Bild des USCT auf der Berechnung von zurückgelegten Wegstrecken des Schalls. Umgekehrt kann aus einem Bildpunkt genau auf einen Samplepunkt in einem A-Scan geschlossen werden, dessen kürzester Abstand zu Sender und Empfänger genau der aus der Signallaufzeit und Schallgeschwindigkeit errechneten Distanz entspricht. Der Benutzer kann anhand der Berechnung erkennen, welches Signal im A-Scan den markierten Bildbereich miterzeugt. Somit kann beispielsweise die Herkunft von Bildartefakten lokalisiert und darauf aufbauend interpretiert werden.

Bei der Zurückrechnung einer ausgewählten ROI auf einen Samplebereich im A-Scan wird die kürzeste Distanz zwischen Sender, ROI und Empfänger als untere Grenze der A-Scan-Selektion benutzt, der weiteste Abstand als obere Grenze.

Zunächst werden - wie bei der Markierung einer Bildselektion im 3D-Modell des USCT - die Eckpunkte der Selektion in das Koordinatensystem des USCT umgerechnet. Anschließend wird die minimale und maximale Entfernung vom aktuell ausgewählten Sender und Empfänger zum Selektions-Rechteck bestimmt. Die maximale Entfernung befindet sich dabei immer an einer Ecke und kann daher

sehr einfach über eine Maximum-Funktion der Abstände aller vier Eckpunkte der Selektion vom Sender und Empfänger bestimmt werden.

Für die minimale Entfernung wird dagegen wesentlich höherer Aufwand nötig, da viele verschiedene Fälle berücksichtigt werden müssen. Gesucht wird dabei der minimale Ellipsoid mit den Positionen des Senders und Empfängers als Brennpunkte, der einen Quader genau in einem Punkt berührt. Zur Lösung des geometrischen Problems wird die Funktion „calcMinDist“ verwendet, die in der USCT-Arbeitsgruppe für andere Software bereits implementiert wurde.

Anhand des maximalen und minimalen Abstandes können basierend auf folgenden Formeln die entsprechenden Samplepunkte im A-Scan berechnet werden:

$$lowerLimit = minDist \cdot timeInterval \cdot sSpeed \tag{6.30}$$

$$upperLimit = maxDist \cdot timeInterval \cdot sSpeed \tag{6.31}$$

wobei $minDist$ der kürzesten Distanz zwischen Sender, ROI und Empfänger entspricht, $maxDist$ der weitesten, $TimeInterval$ dem Zeitintervall zwischen zwei Abtastpunkten und $sSpeed$ der Schallgeschwindigkeit.

Die beiden berechneten Grenzen werden in der A-Scan-Oberfläche als neue Selektion gesetzt und in den geladenen A-Scan eingezeichnet. Anhand der Checkbox „Calc A-Scan“ in der Oberfläche für rekonstruierte Bilder kann gewählt werden, ob die Zurückrechnung automatisch bei einer neuen Markierung im Bild angestoßen wird. In der Callback-Funktion der Checkbox wird dabei ein Flag gesetzt, welches bei der Auswahl einer Region im rekonstruierten Bild berücksichtigt wird.

6.4.3 Zu einer Bildmarkierung beitragende A-Scans

Diese Funktion berechnet anhand einer markierten ROI im Bild diejenigen A-Scans, die am meisten zum Aufbau dieser Region dienten. Sie wurde als Funktion in der Schnittstelle für Bildfunktionen implementiert. Zur Ausgabe der Daten wird das Listenfeld für beliebige Sender-Empfänger-Kombinationen verwendet.

Anhand dieser Funktion erhält der Benutzer Informationen, welche A-Scans maßgeblich zu einer Region im rekonstruierten Bild beigetragen haben. Damit ist es beispielsweise möglich, Aussagen über die Herkunft von unbekannten Bildartefakten zu treffen.

Zunächst holt sich die Funktion Informationen über alle markierten Sender und Empfänger und berücksichtigt dabei das ausgewählte Eingabeobjekt (A-Scan-Listenfelder oder beliebige Sender-Empfänger-Kombinationen).

Anschließend werden die A-Scans zu den entsprechenden Sendern und Empfängern geladen und in Matrizen geschrieben, die darauf folgend durchlaufen werden. Für jeden A-Scan wird wie in 6.4.2 beschrieben auf das Intervall zurückgerechnet, das zum Aufbau der markierten Bildregion beigetragen hat. In diesem wird anschließend die maximale Amplitude bestimmt. Das Ergebnis wird

zusammen mit den Sender- und Empfänger-Nummern in einer neuen Matrix gespeichert:

maxAmplitude	**SL**	**SN**	**RL**	**RN**
:	:	:	:	:

Tabelle 6.2: Datenstruktur bei der Speicherung der maximalen Amplituden eines Bereichs im A-Scan

Anschließend wird eine Sortierung nach der ersten Spalte vollzogen.

Über den Parameter „inputParams.numberCalculated“ wird festgelegt wie viele der berechneten Sender-Empfänger-Kombinationen angezeigt werden sollen. Aus den Datensätzen mit den höchsten Amplituden wird der String mit den Sender-Empfänger-Kombinationen für das Listenfeld generiert und die UserData-Variable aktualisiert.

Da eine Berechnung in Abhängigkeit von der Anzahl der ausgewählten Sender-Empfänger-Kombinationen als Eingabedaten einige Zeit in Anspruch nehmen kann, wird dem Benutzer eine Fortschrittsanzeige eingeblendet.

6.4.4 Berechnung von Rekonstruktionsellipsen

Den entgegengesetzten Weg zur Zurückrechnung auf A-Scanbereiche geht die Funktion „calculateEllipse“, die aus einer Markierung in einem A-Scan eine Ellipse in der Schicht des aktuell angezeigten rekonstruierten Bildes berechnet und diese darin anzeigt. Das Prinzip entspricht dem zur Rekonstruktion eines Bildes verwendeten Prinzip wie in Abschnitt 4.1.4 beschrieben.

Die Funktion ermöglicht dem Benutzer, Informationen über die Verarbeitung der Signale zu einem Bild zu erlangen. Bestimmt werden kann damit, in welche Bildbereiche die Signale eines A-Scans eingezeichnet werden.

Zur Berechnung der Ellipse wird die Funktion „addsig2vol“ aus der Rekonstruktionssoftware verwendet. Als Eingabeparameter erhält sie ein Datenarray, wobei dies einer 1×3000-Matrix entspricht, welche an den Stellen zwischen der linken und rechten Markierung im A-Scan mit dem Wert 1 gefüllt ist, sonst mit 0. Ebenso benötigt die Funktion den Startpunkt des rekonstruierten Bildes, die Position des Sende- und Empfangselements des geladenen A-Scans, die Schallgeschwindigkeit, die physikalische Auflösung des Bildes, das Zeitintervall zwischen zwei Abtastpunkten sowie die Größe des Bildes.

Die Funktion erzeugt eine Matrix, in der an den Stellen, an denen die Ellipse liegt, die Werte auf 1 gesetzt werden. Andere Stellen bleiben 0. Diese Matrix repräsentiert das Bild der berechneten Ellipse zum zugehörigen ausgewählten A-Scan-Bereich (Abbildung 6.17).

Für das Einzeichnen der Ellipse in das Bild kann der Benutzer zwischen zwei Methoden (Abbildung 6.18) wählen. Standardmäßig wird das erhaltene Ellipsen-

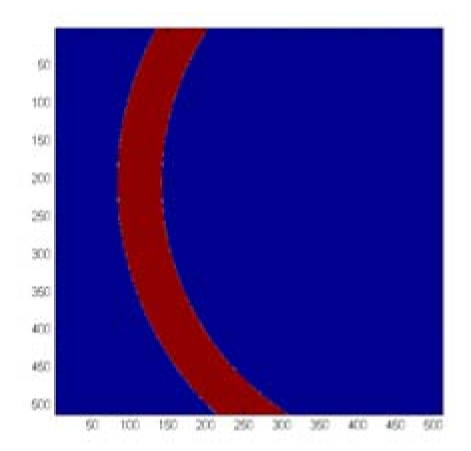


Abbildung 6.17: Das von der Funktion addsig2vol erzeugte Bild zeigt die berechnete Ellipse

Bild auf die Größe der Colormap des rekonstruierten Bildes skaliert und anschließend auf das rekonstruierte Bild addiert. Das neu berechnete Bild wird dann im Anzeigefenster der Bilder-Oberfläche dargestellt.

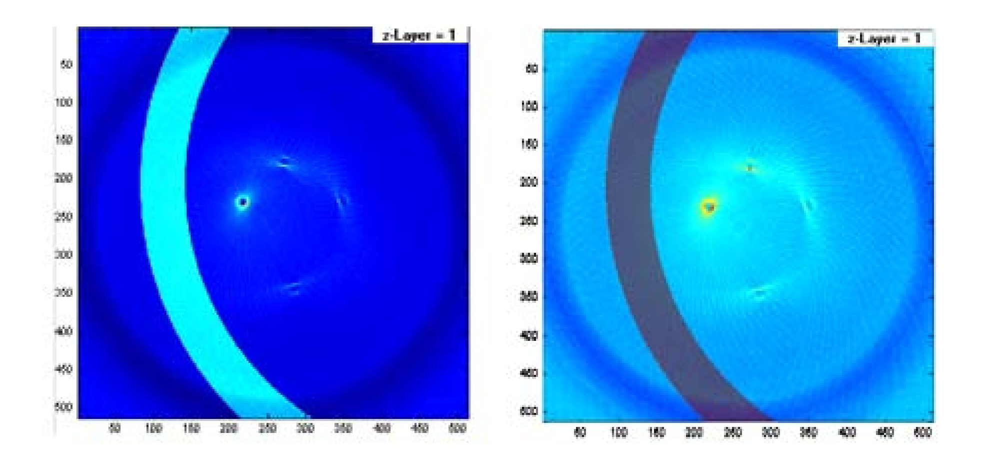


Abbildung 6.18: Anzeige der Rekonstruktionsellipsen durch Addieren (links) und halbtransparente Überlagerung (rechts)

Die zweite Methode ist die halbtransparente Überlagerung der beiden Bilder, wofür OpenGL als Renderer der Oberfläche benötigt wird. Dabei werden Elemente, die zur Ellipse gehören mit einem Transparenzwert von 50% belegt, nicht zur Ellipse gehörige Elemente, die den Wert 0 aufweisen werden zu 100% transparent dargestellt. Im Bild-Objekt muss das Attribut „AlphaData“ mit dem generierten Transparenzbild gefüllt werden. Ein Problem der Darstellung mit OpenGL sind Darstellungsfehler unter MATLAB 6.1, wodurch teilweise die Farben der

Oberfläche verfälscht werden. Deshalb kann die Darstellungsmethode in den benutzerdefinierten Einstellungen der Software geändert werden.

Der Aufruf der Funktion zum Berechnen und Einzeichnen der Rekonstruktionsellipse erfolgt durch die „Calculate Ellipse“-Schaltfläche in der A-Scan-Oberfläche. Des Weiteren verfügt die Oberfläche über eine Checkbox „Auto Calc“, die ein Flag setzt, welches bei der Änderung einer Selektion im A-Scan sowie der Änderung der Temperatur und des dargestellten Schnittbildes abgefragt wird. Ist das Flag gesetzt, wird die Berechnung bereits ohne Klick auf die Schaltfläche ausgeführt.

6.5 Start-Tool

Der Start der Software erfolgt über die Funktion „DiagToolStarter“. Diese öffnet einen Dialog, in dem anhand eines mehrfach aktualisierten Textfeldes der Ladefortschritt der Software angezeigt wird.

Der Ladefortschritt der Software wird dem Benutzer angezeigt, da die Initialisierung der Oberflächen einige Rechenzeit in Anspruch nimmt. Vor allem das rekursive Einbinden der Pfade erweist sich als zeitintensive Operation.



Abbildung 6.19: Startfenster der Software mit Ladeanzeige

Das Start-Tool öffnet zunächst die drei Hauptfenster, wobei die Benutzeroberflächen unsichtbar gehalten werden. Dabei werden die jeweiligen initialen Einstellungen der Funktionen vorgenommen. Anschließend werden die Fenster unter Berücksichtigung der Bildschirmauflösung des Benutzers nebeneinander angeordnet.

In einem weiteren Schritt wird die 3D- und 2D-Ansicht der USCT-Oberfläche durch den Aufruf der jeweiligen Subfunktionen geladen. Abschließend wird das Sichtbarkeitsattribut der Oberflächen gesetzt und die DiagToolStarter-Oberfläche geschlossen, womit die Software fertig geladen ist und nun verwendet werden kann. Die Oberfläche stellt sich nach dem Start der Software wie in Abbildung 6.20 dar.

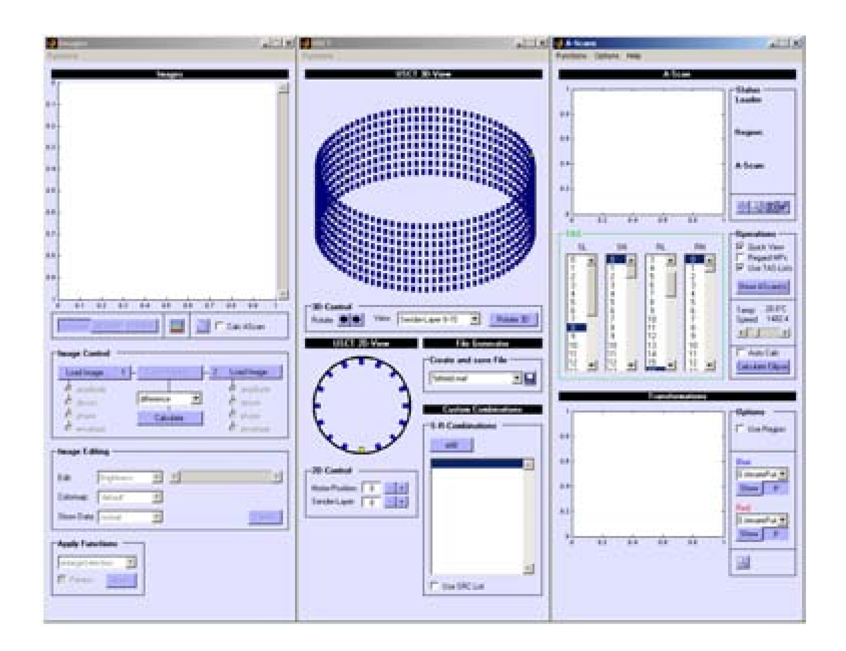


Abbildung 6.20: Anordnung der Benutzeroberflächen nach dem Starten der Software

6.6 Implementierte Plugins

Beispielhaft wurden einige Plugins der Schnittstelle für eigenständige Funktionen implementiert. Deren Funktionsweise sei hier kurz vorgestellt.

6.6.1 Visualisierung der Geometrie-Dateien

Für die Visualisierung der Geometrie-Dateien wird eine neue Oberfläche (Abbildung 6.21) geöffnet, in der durch zwei Textfelder der Pfad zu den Geometrie-Dateien angegeben werden kann. Neben der Eintragung des Pfades „per Hand“ ist es möglich, die Dateien anhand eines Öffnen-Dialoges auszuwählen.

Durch das Betätigen der Schaltfläche „Visualize“ werden die in den Geometrie-Dateien gespeicherten Koordinaten der Sender und Empfänger ausgelesen und mit dem MATLAB-Befehl „plot3“ in das Axes-Objekt der Oberfläche geplottet. Sender werden als rote Punkte dargestellt, Empfänger als grüne Punkte.

6.6.2 Transducer Check

Der Transducer Check dient der Überprüfung der Sende- und Empfangselemente zur Detektion fehlerhafter Ultraschallwandler und nicht verwertbarer A-Scans. Die Oberfläche ist aufgeteilt in einen Bereich zur Überprüfung der Sender und einen Bereich zur Überprüfung der Empfänger (Abbildung 6.22). Gearbeitet wird auf dem aktuell in der Software eingestellten Experimentverzeichnis. Dabei sollte

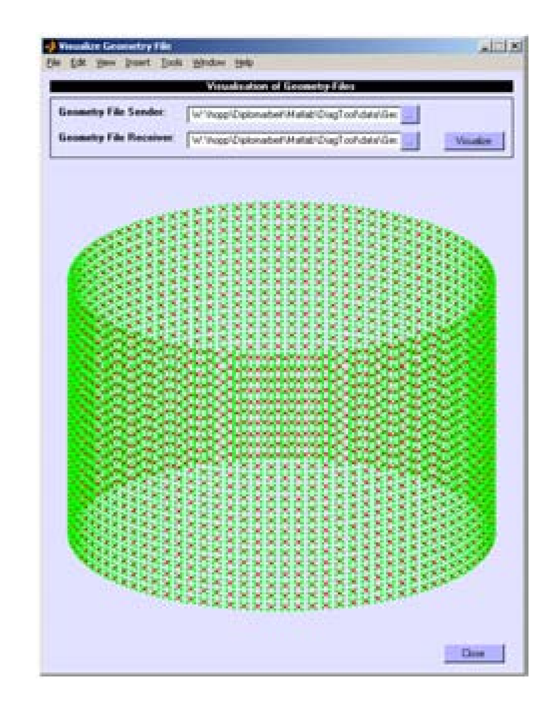


Abbildung 6.21: Oberfläche der Funktion zur Visualisierung der Geometrie-Dateien

beachtet werden, dass ein Transducer Check am sinnvollsten auf einer Leermessung arbeitet, d.h. ein Experiment, bei dem sich außer Wasser kein Objekt im USCT-Zylinder befindet.

Das Testverfahren summiert die zu einem Sender gehörigen Transmissionssignale[5] innerhalb der A-Scans aller Empfänger bzw. die zu einem Empfänger gehörigen Transmissionssignale aller Sender auf und bestimmt somit einen charakteristischen A-Scan des jeweiligen Transducers.

In einer Visualisierungsfunktion die nach Abarbeitung des Transducer-Checks ausgeführt werden kann, werden die maximalen Amplituden der charakteristischen A-Scans für jeden Sender bzw. Empfänger berechnet und als Farbwerte in einem Anzeigefeld dargestellt. Die x-Achse enstpricht dabei den Sende- bzw. Empfangslayern, die y-Achse stellt die Sende- bzw. Empfangsnummern dar.

Darüber hinaus kann auf der Ergebnismenge eine Funktion zur Bestimmung des Signal-Rausch-Verhältnisses angewendet werden. Über die Abfrage eines Schwellwertes werden diejenigen Sender bzw. Empfänger in einem Listenfeld aufgelistet, die diesen Schwellwert nicht erreichen.

Das Ergebnis eines Transducerchecks kann im MATLAB-Datenformat abgespeichert werden, so dass die zeitaufwendige Abarbeitung des Tests für verschiedene weitere Funktionen genutzt werden kann.

[5] Mit Transmissionssignal wird das Signal innerhalb des A-Scans bezeichnet, das auf direktem Weg von Sender zu Empfänger gelangt ist.

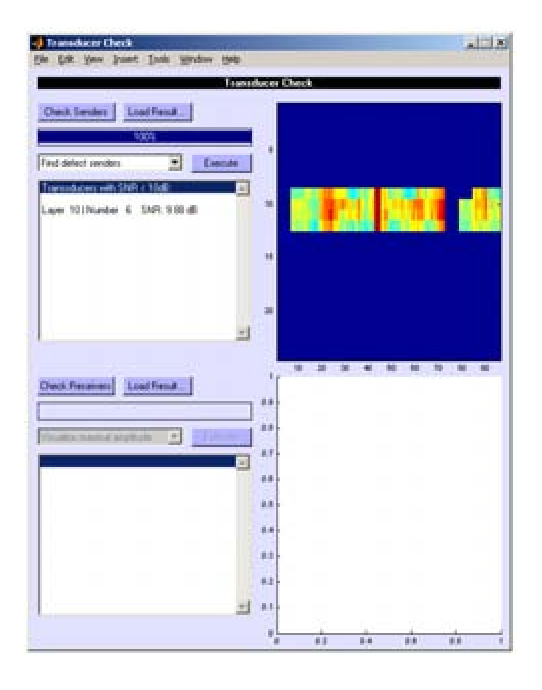


Abbildung 6.22: Oberfläche des Transducer Checks

6.6.3 Visualisierung des Anregungspulses

Wie in Kapitel 4 beschrieben, wird der Anregungspuls in einer Datei „codedExcitation.txt“ als Aneinanderreihung von Zahlenwerten gespeichert. Diese wird durch die Funktion „showCodedExcitation“ anhand des in der Software eingestellten Experimentverzeichnisses ausgelesen und die Werte in ein MATLAB-Array geschrieben, welches anschließend in einem neuen Fenster geplottet wird (Abbildung 6.23).

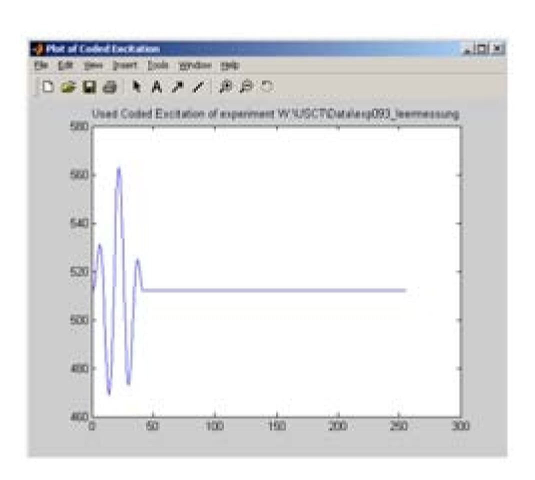


Abbildung 6.23: Visualisierung des bei der Messung verwendeten Anregungspulses

6.7 Dokumentation der Software

Zur Dokumentation der Software wurde ein Benutzerhandbuch erstellt, in dem alle Funktionalitäten aufgelistet und erklärt werden. Dieses ist im Anhang A zu finden. Eingegangen wird darin auch auf die Schnittstellen, über die der Software neue Funktionen hinzugefügt werden können. Bei der Schreibweise wurde auf die Zielgruppe der Software geachtet, so dass von einem Vorwissen im Zusammenhang mit dem Ultraschall-Computertomographen ausgegangen werden kann. Das Benutzerhandbuch ist auch als PDF-Datei verfügbar und wird zusätzlich als Hilfe-System innerhalb der Software angeboten. Letzteres kann in der A-Scan-Oberfläche über „Help“ > „User Manual“ aufgerufen werden. Der in LaTeX verfasste Text wurde dabei in HTML-Seiten exportiert, welche von MATLAB im Hilfe-Dialog (Abbildung 6.24) ausgegeben werden können.



Abbildung 6.24: Hilfe-Dialog der Software

Die Funktionen und Subfunktionen, die in Quelltextdateien gespeichert die Software ergeben, werden in der Funktionsreferenz beschrieben, welche mit Hilfe des Tools „m2html“[6] von *Guillaume Flandin* generiert wurde. Dieses Skript für MATLAB ist über die MATLAB File Exchange-Seite erhältlich und bietet viele Einstellmöglichkeiten. Die erstellten HTML-Dateien beinhalten zunächst eine kurze Beschreibung der Funktion, die aus den im Quelltext vorhandenen Kommentaren extrahiert wird, gefolgt von einer Auflistung aller aufgerufener und auf-

[6] m2html: Toolbox zum automatischen Generieren einer Dokumentation als HTML-Seiten aus .m-Dateien heraus. Quelle: MATLAB File Exchange, `http://www.mathworks.com/matlabcentral/fileexchange`, Stand: 12.09.2006

rufenden Funktionen. Des Weiteren wird ein Überblick über die Subfunktionen geboten.

Die Zusammenhänge zwischen den Dateien werden zusätzlich in einem erstellten Diagramm aufgezeigt, welches im Anhang C zu finden ist.

Eine Hilfestellung für die Programmierung eigener Plugins bieten die erstellten Vorlagen, die dem Entwickler die Struktur der jeweiligen Schnittstelle vorgeben und in die er seinen Quelltext einfügen kann.

Kapitel 7

Ergebnisse

Erstellt wurde im Rahmen der Diplomarbeit die im Detail beschriebene Software, die den Namen „DiagTool" erhält. Die folgenden funktionalen Anforderungen aus Kapitel 2 wurden dabei umgesetzt:

Die zuvor sehr aufwendige Anzeige der Rohdaten kann anhand der Oberfläche für A-Scans stark vereinfacht werden. Das Laden und Plotten der A-Scans wird unter der Angabe des Experimentverzeichnisses sowie des Senderlayers, der Sendernummer, des Empfängerlayers und der Empfängernummer über grafische Bedienelemente ermöglicht. Für die Anpassbarkeit an Neuerungen im Datenformat ist die Funktion modular austauschbar gestaltet. Einen schnelleren Überblick über mehrere A-Scans, sowie eine Vergleichsmöglichkeit zwischen A-Scans verschiedener Sender-Empfänger-Kombinationen bietet die geschaffene Funktion zur Anzeige mehrerer A-Scans gleichzeitig.

Die Abspeicherung der Rohdaten per Benutzeroberfläche wurde sowohl für die Einzelanzeige als auch die Anzeige mehrerer A-Scans ermöglicht. Dem Benutzer hilft ein „Speichern-unter"-Dialog zur Auswahl des Dateipfades. Ein Export der geplotteten Signaldaten in die Grafikformate JPEG und PNG ist ebenfalls möglich. Der Aufwand zur Erlangung von Messungs- und A-Scan-Informationen wurde durch die Aufbereitung der Messungsinformationen in einem Dialog auf einen Klick reduziert.

Entwickelt wurde ein Konzept zur Anzeige von Volumenbildern in der Benutzeroberfläche, welches die bisher in der Abteilung verwendete manuelle Methodik ablöst. Zur manuellen oder automatisierten Weiterverarbeitung außerhalb der erstellten Software können die Bilddaten sowohl im MATLAB-Datenformat wie auch als JPEG-Bild abgespeichert werden.

Die Möglichkeit zum Vergleich zweier Bilder, die beispielsweise mit unterschiedlichen Parameterkonfigurationen rekonstruiert wurden, ist durch die Implementierung zweier Bildkanäle in der Rekonstruktionsbilder-Oberfläche gegeben. Informationen zum Bild können ohne großen Aufwand innerhalb von Tooltips eingesehen werden. Die Anpassbarkeit des Bildbereiches auf neue Rekonstruktionsverfahren begründet sich in der beliebigen Austauschbarkeit der in der Bilddatei

gesuchten Variablen.

Vereinigungen von Bildern erlauben zusätzliche Möglichkeiten zur Analyse unterschiedlich parametrisierter Bildrekonstruktionen. Diese Funktion wird außerdem über eine Schnittstelle realisiert, die es dem Benutzer erlaubt beliebige Berechnungen zwischen zwei Bilddatensätzen durchzuführen und diese integriert in der Oberfläche anzuzeigen und weiter zu verwenden.

Verbesserungen in der Bildqualität werden sich durch die Verwendung von Bildbearbeitungsfunktionen erhofft. Die Verwendung solcher ist in der erstellten Software beispielhaft durch die Anpassung der Helligkeit gewährleistet. Der Funktionsumfang kann durch die Anordnung und Art der Bedienelemente leicht erweitert werden. In der medizinischen Bildverarbeitung wird mit unterschiedlichen Farbskalen gearbeitet. Die Veränderung der Colormap per einfachem Mausklick ermöglicht die Anzeige der Bilder in unterschiedlichen Farbdarstellungen. Eine häufig gewünschte logarithmische Darstellung der Bilddaten ist ebenso möglich.

Eine engere Bindung zwischen dem realen USCT-Prototypen und den daraus gewonnenen Daten verschafft ein interaktives, dynamisch generiertes 3D-Modell. Es ermöglicht die einfache Auswahl von Sendern und Empfängern über anklickbare Elemente. Verschiedene Markierungsmuster, erreichbar über Kontextmenüs, erleichtern und beschleunigen die Auswahl. Zusätzlich bietet das 3D-Modell mit einer programmierbaren linken Maustaste eine völlig neue Möglichkeit, gleiche Aktionen schnell hintereinander ausführen zu können.

Zur bisher weitgehend fehlenden Verdeutlichung der Position eines rekonstruierten Bildes im USCT-Zylinder werden Auswahlbereiche aus einem Bild in das 3D-Modell eingezeichnet, was eine wesentlich einfachere Zuordnung des rekonstruierten Bildes zum vermessenen Objekt ermöglicht.

Während das 3D-Modell für großflächige Auswahlen und zur Visualisierung sehr gut geeignet ist, bietet ein zusätzliches 2D-Modell die Möglichkeit Sender-Empfänger-Kombinationen präzise auszuwählen.

Die aufwendige händische Erstellung von Parameterdateien für die Datenaufnahme- und Rekonstruktionssoftware ersetzt die erstellte Software mit Funktionen zur automatischen Generierung solcher Dateien anhand von zuvor in der Oberfläche markierter Sender und Empfänger.

Die Visualisierung der wichtigen Verbindung zwischen den Rohdaten und den rekonstruierten Bildern, die vor Einführung der Software nicht vorhanden war, wurde über die Funktionen zur Berechnung von Rekonstruktionsellipsen, Zurückrechnung von Bildpunkten auf A-Scan-Bereiche sowie die Berechnung der am meisten zu einem Bildpunkt beitragenden A-Scans umgesetzt. Sie ermöglicht erstmals die genaue Analyse, wie Teile eines rekonstruierten Bildes mit der entsprechenden Teilmenge von A-Scans zusammenhängen. Die Ausführung erfolgt nach einfachen Mausklicks zum Setzen von Regionen im Bild bzw. in A-Scans.

Die Erweiterbarkeit der Software wird durch die erstellten Schnittstellen gewährleistet, über die sehr einfach eigene Funktionen für A-Scan-

Transformationen, Bildvereinigungen, Funktionen auf Bildern bzw. Bildausschnitten eingebunden werden können. Entstanden ist damit ein Plugin-Konzept, das zu Testzwecken für die Benutzung bereits freigegeben wurde. Bei der Einbindung der Funktion „estimateFWHM“ in der Schnittstelle für Funktionen auf Bildausschnitten, sowie der Funktion „DetectPulses“ in der Schnittstelle für A-Scan-Transformationen, war zu erkennen, dass bei vorhandener Funktionalität die Integration einer Funktion in wenigen Minuten von statten gehen kann. Eine zusätzliche beliebige Erweiterungsmöglichkeit wurde mit der Schnittstelle zur Einbindung von Funktionen in die Menüleisten der Oberflächen geschaffen. Damit kann grundsätzlich jede MATLAB-Anwendung in die Software eingebunden werden. Über in der Funktionsreferenz aufgeführte Funktionen ist auch der Zugriff auf Ein- und Ausgabeelemente der Software möglich.

Alle Funktionen sind in einer dreiteiligen Benutzeroberfläche verfügbar, wodurch die unterschiedlichen Sichtweisen auf die Daten sowohl isoliert als auch vereint gesehen werden können.

Bei der Auswahl von A-Scans herrscht Konsistenz zwischen den verschiedenen Auswahlelementen. Beispielsweise bewirkt eine Markierung in den Listenfeldern automatisch die Aktualisierung der 3D- und 2D-Ansicht. Ein zurückgerechneter A-Scan-Bereich beispielsweise, kann für die Ausführung von Transformationen gesetzt werden.

Durch die Verwendung von Kontextmenüs konnte die Komplexität der Oberfläche reduziert werden. Trotzdem können alle Funktionen über maximal vier Mausklicks erreicht werden.

Bei der Gestaltung der Oberflächen wurde auf Konsistenz geachtet, wodurch einige Prinzipien des guten GUI-Designs eingehalten werden konnten. So wurden die Farben von Fenstern und Schaltflächen durchweg gleich gewählt, ebenso wie die Bedeutung der Texte der Schaltflächen. Die Benutzeroberfläche wurde mit englischen Bezeichnungen versehen, so dass das Verständnis für nicht deutschsprachige Mitarbeiter der Abteilung gewährleistet ist. Ein fehlendes Package-Konzept wird durch eine Strukturierung der Quelltextdateien in Ordnern und Unterordnern ersetzt.

Die Lauffähigkeit der Software wurde unter der MATLAB-Version 6.1 sowie 7.1 überprüft, wobei festzustellen ist, dass einige Befehle der Version 6.1 unter MATLAB 7.1 als inkorrekt erkannt wurden und aufgrund dessen auf synonyme Befehle zurückgegriffen wurde, die beide Versionen identisch interpretieren.

Kapitel 8

Diskussion und Ausblick

Vorgestellt wurde in dieser Diplomarbeit ein Konzept sowie dessen Umsetzung für eine Software, die vor allem darauf ausgerichtet ist, den wissenschaftlichen Mitarbeitern der USCT-Arbeitsgruppe ein Werkzeug zum schnelleren Verständnis von Zusammenhängen bereitzustellen. Vor dem Einsatz der Software arbeitete ein Großteil der in der Arbeitsgruppe entwickelten Funktionen entweder auf den Rohdaten oder auf den rekonstruierten Bildern. Die Auswahl von Teildatensätzen gestaltete sich aufgrund der notwendigen Angabe von bis zu acht Zahlenwerten recht schwierig, ebenso wie das Herstellen von Verbindungen zur Rekonstruktion oder bestimmten Bereichen in einem Volumen. Generell war für die Analyse der Daten ein beträchtliches Hintergrundwissen notwendig, so beispielsweise die Geometrie des Ultraschall-Computertomographen.

Durch den Einsatz der implementierten Software ergeben sich Möglichkeiten, die Zusammenhänge zwischen verschiedenen zur Bildqualität beitragenden Faktoren schneller und unter weniger Aufwand zu erfassen. Dies begründet sich vor allem in der Visualisierung von Informationen und Querverbindungen, die für den Benutzer eine enorme Erleichterung darstellen können. Die Visualisierung stützt sich unter anderem auf einem interaktiven 3D-Modell des USCT, das aus beliebigen Geometrie-Dateien dynamisch generiert wird. Es kann somit auch für zukünftige geometrische Anordnungen der Sender und Empfänger-Elemente benutzt werden. „DiagTool“ bildet damit einen weiteren Schritt in der ständigen Verbesserung des USCT-Prototypen auf dem Weg zu einem marktreifen System.

Einer der wichtigsten Anforderungen war bei der Entwicklung der Software von Beginn an die Erweiterbarkeit. Dafür wurde ein neuartiges Schnittstellen-Konzept geschaffen, das die Einbindung von vorhandenen Funktionen in sehr kurzer Zeit erlaubt.

MATLAB als Programmiersprache erlaubte eine hohe Geschwindigkeit in der Entwicklung. Die Gestaltung der Oberflächen mit dem Graphical User Interface Development Environment erwies sich als sehr gutes Hilfsmittel. Darüber hinaus vereinfachte MATLAB durch die durchgehend vorhandene Verwendung von Matrizen als Datenstrukturen die Entwicklung von Algorithmen und die Speicherung

der Daten der Software.

Als eines der größten Probleme in der Entwicklung der Software lies sich jedoch die Vielzahl von unterschiedlichen Datenstrukturen und Koordinatensysteme ausmachen. Allein die Nummerierung der Sender und Empfänger lässt sich nach fünf unterschiedlichen Systemen vollziehen, die jeweils für andere Funktionen benötigt werden. Das Mapping dazwischen erwies sich als recht aufwendig. Eine weitere Schwierigkeit ergab sich aus der Fülle der Funktionen, die in die Software integriert werden sollten. Diese erhöht sich nochmals durch die Allgemeinhaltung, so dass die Ein- und Ausgaben von Funktionen geändert werden können - wie es beispielsweise bei der Anzeige von A-Scans durch Auswahlen in zwei unterschiedlichen Listenfeldern benötigt wurde. Hindernisse gab es zusätzlich bezüglich der Änderbarkeit von Parametern, bei denen sich schließlich für die aufwendige aber sehr benutzerfreundliche Variante der Eingabe über dynamisch generierte Dialoge entschieden wurde. Aufgrund der nur eingeschränkt gegebenen Objektorientierung der Programmierumgebung MATLAB ergaben sich Probleme, den Überblick über sehr viele unterschiedliche komplexe Strukturen und oberflächenübergreifende Zugriffe auf Bedienelemente zu bewahren. Der Einsatz einer klaren Strukturierung in Ordnern und Unterordnern sowie die Zuordnung von globalen Variablen zu jeder Oberfläche konnten helfen die Komplexität der Software zu reduzieren.

Der aktuelle Stand der Software wird mittlerweile in der USCT-Abteilung eingesetzt. Über ein Subversion-System können die Mitarbeiter immer mit der aktuellsten Version arbeiten. Aufgrund der kurzen Testphase, sollen auftretende Probleme in nächster Zeit sukzessive weiter entfernt werden.

Während der Entwicklung ergaben sich zusätzlich zu den anfangs formulierten Anforderungen Ideen und Wünsche bezüglich der Funktionalität und der Bedienung. Teilweise wurden diese bereits in der nun fertiggestellten Version umgesetzt - beispielsweise sei die Implementierung des Transducer-Checks oder die Visualisierung des Anregungspulses genannt. Weitere sollen in Zukunft in die Software integriert werden:

- Generierung von synthetischen A-Scan-Daten.
- Ausweitung der Auswahlmöglichkeiten für A-Scan-Bereiche auf die Anzeige mehrerer A-Scans. Hierfür wurde bereits eine Machbarkeitsuntersuchung mit positivem Ergebnis durchgeführt.
- Überlagerung von unterschiedlichen A-Scans.
- Generierung des Anregungspulses und Abspeichern in einer Datei entsprechenden Formates.
- Implementierung der Resize-Funktionen. Damit ist es möglich die Oberflächen an verschiedene Bildschirmauflösungen anzupassen.
- Anbieten weiterer Export-Formate für Grafiken. Auch hier könnte evtl. ein Schnittstellenkonzept zum Einsatz kommen.

- Hinzufügen von weiteren Bildbearbeitungsfunktionen.
- Anpassung des 3D-Modells, so dass die ausgewählten Motorpositionen visualisiert werden. Dies könnte evtl. über die Änderung der Farbe der Voxel erfolgen.
- Implementierung zusätzlicher Auswahlmuster für das 3D-Modell sowie das Listenfeld für beliebige Sender-Empfänger-Kombinationen.
- 3D-Auswahl von Bereichen im USCT-Modell.
- Starten der Rekonstruktionssoftware über die Benutzeroberfläche. Hierfür sind jedoch zunächst Anpassung an der Rekonstruktionssoftware notwendig.

Des Weiteren soll die Erweiterung der Software mit dem Einbinden vorhandener Funktionen durch die Mitarbeiter vorangetrieben werden. Auch hier hilft das Subversion-System den Benutzern immer den aktuellen Funktionsumfang bereitzustellen.

Eine weiterreichende Änderung wäre das Kompilieren des Quellcodes mit dem MATLAB-Compiler zu einer eigenständigen Software. Diese Zusatzsoftware kann den MATLAB-Quellcode in kompilierbaren C- bzw. C++-Quelltext umwandeln, wodurch sich Geschwindigkeitsvorteile bei der Ausführung sowie eine Unabhängigkeit von MATLAB ergeben können. Nachteilig ist jedoch, dass hierdurch die einfache Erweiterbarkeit der Software in nicht absehbarem Maße beeinträchtigt werden würde.

Für eine Anbindung an eine Datenbank bietet MATLAB in seinem großen Funktionsumfang ebenfalls Möglichkeiten, so dass bei einer Verlagerung der Experimentdaten in eine Datenbank möglicherweise eine bessere Strukturierung erreicht werden könnte.

Ein völlig neuer Ansatz ergab sich bei Überlegungen bezüglich der Aneinanderreihung von verschiedenen Operationen. Um dies zu ermöglichen bietet sich das Konzept einer Komponentensoftware an. Verschiedene Funktionscontainer könnten über definierte Schnittstelle beliebig miteinander verbunden werden.

Literaturverzeichnis

[AAA04] ABDULLA, Dr. Abdulla I. ; ASHI, Reefat Yousef A. ; AMERI, Ahmed A.: *Introduction to Graphical User Interface (GUI) Matlab 6.5.* 2004. – United Arab Emirates University

[GHJL05] GIERSIEPEN, Klaus ; HEITMANN, Cornelia ; JANHSEN, Katrin ; LANGE, Cornelia: *Gesundheitsberichterstattung des Bundes - Heft 25 - Brustkrebs.* Robert Koch-Institut, 2005

[Har03] HARTMANN, Peter: *Mathematik für Informatiker.* Vieweg Verlag, 2. Auflage, 2003

[Hob95] HOBART, James: Principles of good GUI Design. In: *Unix Review, vol. 13, num 10* (1995), S. 37–46

[Hop05] HOPP, Torsten: *Schrittmotoransteuerung für 3D Ultraschall Computer-Tomographie am FZK*, Berufsakademie Mannheim, Forschungszentrum Karlsruhe - Institut für Prozessdatenverarbeitung und Elektronik, Praxisbericht, 2005

[Kar04] KARRENBERG, Ulrich: *Signale Prozesse Systeme - eine multimediale und interaktive Einführung in die Signalverarbeitung.* Springer, 2004

[MR05] MUELLER, Tim ; RUITER, Nicole. *Specification of the Software-View on the 3D USCT Geometry and appropriate Data Management.* 2005

[Nat04] NATIONAL PHYSICAL LABORYTORY: *Technical Guides - Speed of Sound in Pure Water.* `http://www.npl.co.uk/acoustics/techguides/soundpurewater/content.html`. 2004. – [Online; Stand 15. August 2006]

[Sko05] SKOGEN, Martha G. R.: *Design Principles for Electronic User Interfaces.* Vortragsfolien. Oktober 2005. – Norwegian University of Science and Technology - Department of Product Design

[The00a] THE MATHWORKS INC.: *Using MATLAB Graphics Version 6.* The Mathworks Inc., 2000

[The00b] THE MATHWORKS INC.: *Using MATLAB Version 6.* The MathWorks Inc., 2000

[The01a] THE MATHWORKS INC.: *Image Processing Toolbox User's Guide Version 3.* The Mathworks Inc., 2001

[The01b] THE MATHWORKS INC.: *MATLAB Compiler User's Guide Version 2.* The Mathworks Inc., 2001

[The02] THE MATHWORKS, INC.: *Creating Graphical User Interfaces - Version 6.* The MathWorks, Inc., 2002

[The05] THE MATHWORKS INC.: *MAT-File Format.* The MathWorks Inc., 2005

[The06] The MathWorks Inc.: *MATLAB 7.3 Produktbeschreibung*. http://www.mathworks.de/products/matlab/description1.html. 2006. – [Online; Stand 11.September 2006]

[Wik06a] Wikipedia: *Magnetresonanztomographie — Wikipedia, Die freie Enzyklopädie*. http://de.wikipedia.org/wiki/Magnetresonanztomographie. 2006. – [Online; Stand 11. September 2006]

[Wik06b] Wikipedia: *Mammographie — Wikipedia, Die freie Enzyklopädie*. http://de.wikipedia.org/wiki/Mammografie. 2006. – [Online; Stand 11. September 2006]

[Wik06c] Wikipedia: *Sonographie — Wikipedia, Die freie Enzyklopädie*. http://de.wikipedia.org/wiki/Sonographie. 2006. – [Online; Stand 11. September 2006]

[Wür00] Würfel, Jan U.: *Dreidimensionale Ultraschalltomographie: Machbarkeitsstudie und Aufbau eines Prototyps*, Institut für Experimentelle Kernphysik (IEKP), Universität Karlsruhe / Hauptabteilung Prozessdatenverarbeitung und Elektronik (HPE), Forschungszentrum Karlsruhe, Diplomarbeit, Dezember 2000

Abkürzungsverzeichnis

μs	Mikrosekunden
2D	zweidimensional
3D	dreidimensional
A-Scan	Amplituden-Scan
ASCII	American Standard Code for Information Interchange
CTR	Control Board
DLL	Dynamic Linked Library
FLT	First Level Triggerboard
FZK	Forschungszentrum Karlsruhe
GB	Gigabyte
GUI	Graphical User Interface
GUIDE	Graphical User Interface Development Environment
HTML	Hypertext Markup Language
IPE	Institut für Prozessdatenverarbeitung und Elektronik
JPEG	Joint Photographic Experts Group
LCD	Liquid Crystal Display
m	Meter
MHz	Megahertz
mm	Millimeter
MRT	Magnet-Resonanz-Tomographie
OpenGL	Open Graphics Library
PBus	Protokoll Bus
PNG	Portable Network Graphics
RGB	Rot Grün Blau
RL	Receiverlayer

RN	Receivernumber
ROI	Region of Interest
SL	Senderlayer
SLT	Second Level Triggerboard
SN	Sendernumber
SVN	Subversion
TAS	Transducer Array System
USCT	Ultraschall-Computertomograph

Abbildungsverzeichnis

Anhang A

Benutzerhandbuch

A.1 Vorbemerkungen

Das DiagTool ist eine in MATLAB geschriebene Software, die es ermöglichen soll, die großen Datenmengen, die bei einer Messung mit dem USCT entstehen, zu durchsuchen, zu analysieren und zu visualisieren.

Hierfür verbindet sie die Bereiche Rohdaten (A-Scans), rekonstruierte Bilder (Images) und bietet zudem mit dem Fenster „USCT“ eine Visualisierungsmöglichkeit des Ultraschall-Computer-Tomographen in 2D und 3D.

A.1.1 Systemanforderungen

Das DiagTool ist in MATLAB implementiert und benötigt deshalb einen Rechner (unabhängig vom Betriebssystem), auf dem The Mathworks Inc. MATLAB ab Version 6.1 installiert ist. Für eine korrekte Darstellung der Benutzeroberfläche wird eine Bildschirmauflösung von 1280 $\times$ 1024 Pixel empfohlen.

A.2 Installation und Starten der Software

Die Software „DiagTool“ liegt als gepacktes ZIP-Archiv vor, in dem alle benötigten Dateien, Daten und Ordnerstrukturen enthalten sind. Zur Installation muss dieses Archiv in einen Ordner auf der Festplatte entpackt werden.

Im obersten Verzeichnis befindet sich eine Batch-Datei (.bat), durch die die Software gestartet werden kann. Sie lädt lediglich die als Standard definierte Version von MATLAB und führt anschließend die Datei „DiagToolStarter.m“ aus. Alternativ zur Batch-Datei kann das Programm auch manuell über die MATLAB-Kommandozeile gestartet werden. Hierfür muss in MATLAB das „Current Directory“ ausgewählt werden, in den das DiagTool entpackt wurde. Darauffolgend kann in die Kommandozeile von MATLAB der Befehl `DiagToolStarter;` eingegeben werden.

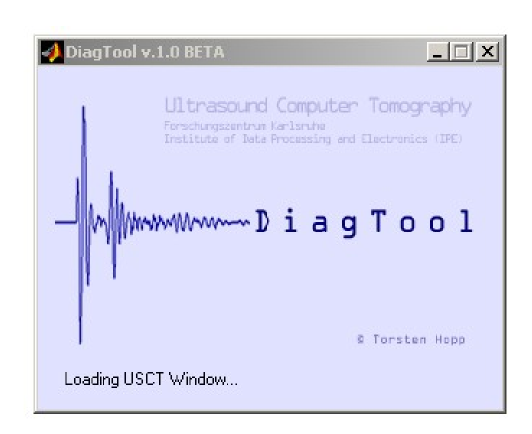


Abbildung A.1: Oberfläche des DiagToolStarter

Der Start der Software sollte möglichst immer über den DiagToolStarter erfolgen, um alle Variablen zu initialisieren. Bei Ausfürhung des DiagToolStarter öffnet sich ein Popup-Fenster, in dem der aktuelle Ladestand angzeigt wird (Abbildung A.1). Nach kurzer Zeit erscheint ein OK-Button, der bestätigt werden muss, um die Initialisierung der USCT-Ansichten vorzunehmen und die Oberfläche (Abbildung A.2) zu öffnen. Während der Laufzeit der Software sollte das Arbeitsverzeichnis von MATLAB nicht verändert werden.

Die Oberfläche ist in drei Bereiche unterteilt: Images, USCT und A-Scans, deren Funktionen im Folgenden jeweils näher beschrieben werden.

A.3 A-Scans

Die Benutzeroberfläche „A-Scans“ (Abbildung A.3) arbeitet auf den Rohdaten einer USCT-Messung.

A.3.1 Allgemeine Einstellungen

Allgemeine Einstellungen finden sich in der Menüleiste unter „Options“ (2) > „Preferences“. Im sich öffnenden Dialog können die aktuellen Parameter eingesehen und verändert werden:

Setzen des Experimentverzeichnisses Im Experimentverzeichnis sind die Rohdaten einer Messung abgelegt. Diese Einstellung ist zwingend erforderlich und bildet die Grundlage für jegliche weitere Funktionen. Alternativ kann das Experimentverzeichnis auch über einen Rechtsklick auf das A-Scan-Anzeigefeld > „Set Experiment Path...“ neu gesetzt werden.

Bezeichnung der Bildrekonstruktions-Arten Die Bildrekonstruktions-Arten (maximal 4) werden als Variablennamen in einem rekonstruierten

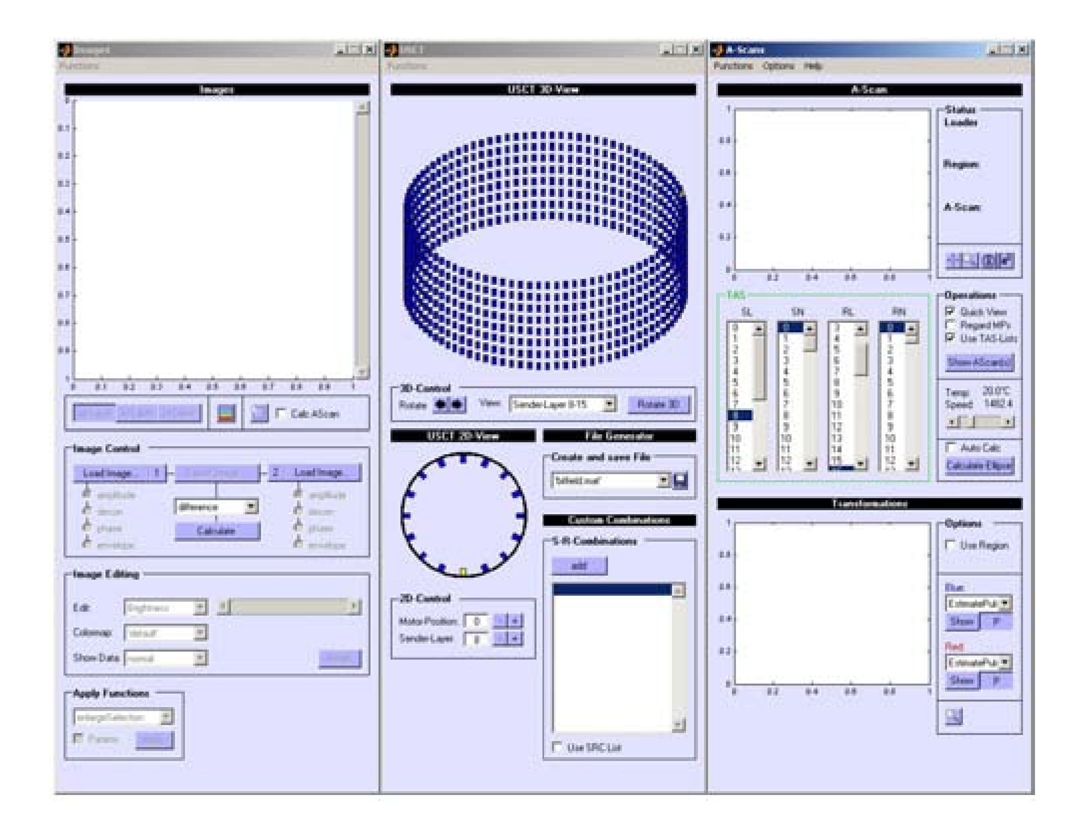


Abbildung A.2: DiagTool-Oberfläche nach dem Starten

Bild verwendet (z.B. 'amplitude', 'decon', 'phase', 'envelope'). Die Bezeichnung kann als einfacher Text in die entsprechenden Felder eingetragen werden.

Standardeinstellung 3D-USCT Bei der 3D-Darstellung des Ultraschall-Computertomographen kann eine initiale Form gewählt werden.

Variablennamen bei rekonstruierten Bildern Innerhalb einer .mat-Datei eines rekonstruierten Bildes stehen mehrere Bildmodi zur Verfügung, die jeweils in einer Variablen gespeichert sind. In den Feldern „Reconstructed Image Variable Names" können die Variablennamen, nach denen gesucht wird, frei gewählt werden. Die Standardeinstellung ist

- amplitude
- decon
- phase
- envelope

Art der Ellipsenrekonstruktion Mit den Radiobuttons im Feld „Ellipse Reconstruction" wird festgelegt wie rekonstruierte Ellipsen in einem angezeigten Bild angezeigt werden. Mit Sum Images wird die Ellipse in das Bild

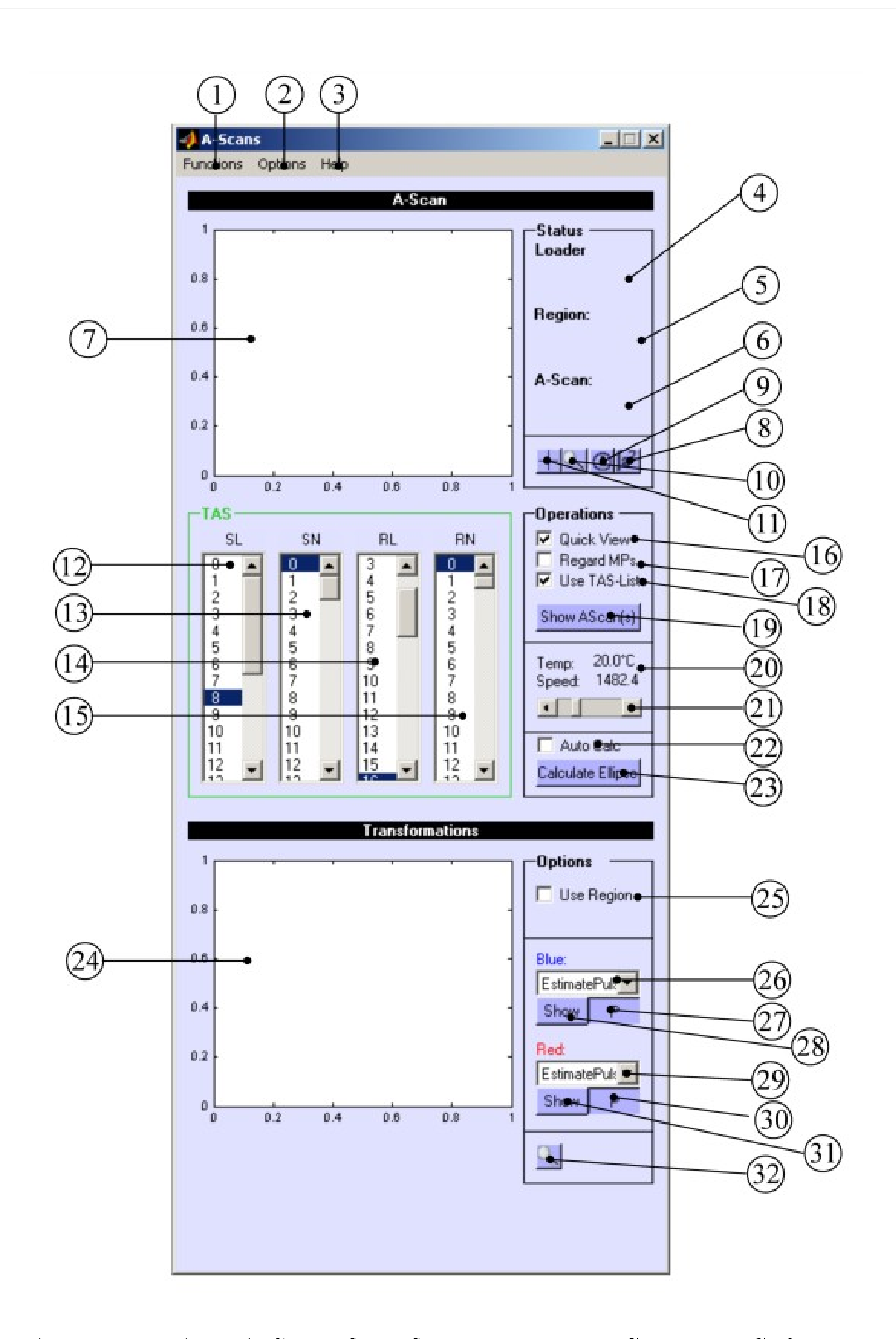


Abbildung A.3: A-Scan-Oberfläche nach dem Start der Software

addiert, mit Transparency wird sie als halbtransparentes Overlay in das Bild gezeichnet. Zu beachten ist, dass die Transparenz Open GL benötigt!

A.3.2 A-Scans auswählen, laden und anzeigen

Zum Laden von A-Scans ist der obere Bereich der Benutzeroberfläche („A-Scan“) vorhanden. Die Auswahl eines bestimmten A-Scans erfolgt anhand der vier Listenfelder SL (für Sender Layer, (12)), SN (für Sender Number, (13)), RL (für Receiver Layer, (14)) und RN (Receiver Number, (15)) im Feld „TAS“. Einzelne Zeilen können per einfachem Mausklick markiert werden, mit dem Halten der Shift-/Strg-Taste bzw. dem Ziehen mit der Maus können mehrere Elemente gleichzeitig ausgewählt werden.

Anzeigen Mit dem Button „Show AScan(s)“ (19) werden entsprechend der Auswahl die A-Scans in den Speicher geladen. Dies geschieht unabhängig von der im Experiment verwendeten Datenstruktur.

Wurde lediglich ein einzelner A-Scan markiert, so wird dieser in das A-Scan-Anzeigefeld (7) geplottet. Wurden dagegen mehrere Kombinationen von Sendern und Empfängern in den Listenfeldern markiert, so werden alle zugehörigen A-Scans in einem neuen Popup-Fenster geplottet. Siehe hierfür auch A.3.4.

Statusinformationen Im Unterpunkt „Loader“ (4) des Statusfeldes werden entsprechend des Ladevorgangs Meldungen ausgegeben, d.h. ob der A-Scan im aktuellen Experimentverzeichnis gefunden wurde und geladen werden konnte oder nicht vorhanden ist. „Region“ (5) gibt die obere und untere Grenze einer Markierung in einem A-Scan an. Unter „A-Scan“ (6) wird die Nummerierung des aktuell geladenen A-Scans angezeigt.

Quick View Die standardmäßig deaktivierte Option Quick View (16) weist die Software an, sofort nach der Auswahl in einer Liste den zugehörigen A-Scan zur entsprechend markierten Sender-Empfänger-Kombination zu laden und anzuzeigen. Diese eignet sich zum schnelleren Durchlaufen der Datensätze.

Regard Motorpositions Als zweite Option wird „Regard MPs“ (17) [1] angeboten. Wird diese aktiviert, so werden nur die Receiver-Nummern in der Liste RN angezeigt, die für den ausgewählten Sender aufgrund der zugehörigen Motorposition Daten enthalten können.

Use TAS-Lists Standardmäßig ist diese Option (18) aktiviert. Sie stellt sicher, dass beim Laden der A-Scans die markierten Sender-Empfänger-Kombinationen der A-Scan-Listen verwendet werden. Durch Deaktivieren dieser Checkbox wird automatisch die Checkbox 'Use SRC-List' gesetzt und durch die Änderung des grünen Rahmens verdeutlicht. Durch den Wechsel werden beim Anzeigen von A-Scans und weiteren Operationen die Sender-Empfänger-Kombinationen aus dem Feld 'Custom Combinations' der USCT-Oberfläche verwendet.

[1] MP = Motorposition

A.3.3 A-Scan-Funktionen

Wurde ein einzelner A-Scan korrekt geladen und wird nun im A-Scan-Anzeigefeld dargestellt, so können weitere Funktionen verwendet werden:

Daten neu abspeichern Durch Rechtsklick auf das A-Scan-Anzeigefeld (7) > „Save A-Scan Data as...“ öffnet sich ein Speichern-unter-Dialog, durch den die gesamten Daten des aktuell angezeigten A-Scans in einer Variablen mit der Bezeichnung „AScan“ als .mat-File mit gewähltem Namen abgespeichert werden.

Bereich markieren Die Schaltfläche mit dem Fadenkreuz (11) ermöglicht es, einen Bereich im angezeigten A-Scan zu selektieren. Es wird ein veränderter Mauszeiger angezeigt mit dem zwei Stellen ausgewählt werden können. Die Auswahl kann mit der Enter-Taste abgebrochen werden. Eine erfolgreiche Bereichsauswahl wird durch zwei grüne Linien im A-Scan an den gewählten Stellen gekennzeichnet. Die Auswahl kann dann anschließend als Grundlage für weitere Berechnungen dienen (siehe z.B. Transformationen). Die Stellen werden zusätzlich als Zahlenwerte im Statusfeld unter „Region“ (5) ausgegeben.

Vergrößerung anzeigen Ein Klick auf das Lupen-Symbol (10) öffnet ein neues Fenster, in dem der aktuelle A-Scan vergrößert dargestellt wird. Alternativ kann zu dessen Aufruf auch ein Rechtsklick auf das A-Scan-Anzeigefeld (7) > „Enlarge“ verwendet werden.

Als Bild exportieren Innerhalb des Vergrößerungsfensters kann durch einen Rechtsklick auf den angezeigten A-Scan > „Save as JPEG...“ bzw > „Save as jpg...“ die derzeitige Darstellung in eine Bilddatei entsprechenden Formates exportiert werden. Zuvor wird ein Speichern-unter-Dialog aufgerufen.

Informationen anzeigen Der Info-Button (9) öffnet ein neues Fenster, in dem Informationen über den aktuellen A-Scan und das zugehörige Experiment angezeigt werden. Alternativ kann für den Aufruf dieser Funktion auch ein Rechtsklick > „Info“ verwendet werden.

Parameter berechnen Durch Klicken auf die Schaltfläche mit den Formel-Symbolen (8) wird ein neues Fenster geöffnet, in dem charakteristische Parameter zum aktuell geladenen A-Scan berechnet und angezeigt werden:

- Durschschnittliche absolute Amplitude (mean)
- Durchschnittliche absolute Amplitude (median)
- Maximale Amplitude und Samplepunkt an der sie auftritt

- Minimale Amplitude und Samplepunkt an der sie auftritt
- Varianz
- Standardabweichung
- Signal-Rausch-Verhältnis

Temperatur und Schallgeschwindigkeit verändern Mit der Bildlaufleiste (21) im Bereich „Operations“ kann die für weitere Berechnungen verwendete Temperatur und Schallgeschwindigkeit (20) angepasst werden.

A.3.4 Mehrere A-Scans anzeigen

Wurden in den Listenfeldern mehrere Sender-Empfänger-Kombinationen ausgewählt, kann durch einen Klick auf den Button „Show A-Scan(s)“ (19) ein neues Fenster geöffnet werden, in dem die entsprechenden A-Scans geplottet angezeigt werden. Zu jedem Plot werden Informationen über Sender- und Empfängernummern angegeben.

Daten abspeichern Mit einem Rechtsklick auf einen A-Scan kann die Option „Save as...“ aufgerufen werden. Es öffnet sich ein Dialog, in dem ausgewählt werden kann, ob ein einzelner oder alle angezeigten A-Scans gespeichert werden sollen. Über einen Speichern-unter-Dialog wird der Speicherort ausgewählt.

Bei einzelnen abgespeicherten A-Scans wird der ausgewählte Dateiname verwendet. Beim gleichzeitigen Abspeichern mehrerer A-Scans wird dem Dateinamen die Sender-Empfänger-Kombination vorangestellt.

A.3.5 Rekonstruktionsellipse berechnen

Mit dem Button „Calculate Ellipse“ (23) kann - sofern ein Bild mit zugehöriger Info-Datei geladen ist - anhand eines Ausschnittes aus dem A-Scan berechnet werden, in welchem Bildbereich dieser Ausschnitt einen Beitrag zum rekonstruierten Bild leistet. Es ist zwingend erforderlich vor der Ausführung der Funktion einen Bereich im A-Scan zu markieren.

Auto Calc Ist die Option „Auto Calc“ (22) gesetzt, wird die Rekonstruktionsellipse bei jeder Neumarkierung im A-Scan, Wechsel der Temperatur/Schallgeschwindigkeit und Wechsel der angezeigten Bildebene sofort berechnet ohne den Button zuvor betätigen zu müssen.

A.3.6 Transformationen

Im unteren Teil der A-Scan-Oberfläche befindet sich der Abschnitt „Transformations“, der es ermöglicht beliebige Funktionen auf den Daten des aktuell geladenen A-Scans auszuführen. Das Ergebnis der Transformation wird dann in das A-Scan-Transformations-Feld (24) geplottet.

Es stehen zwei identisch aufgebaute Transformations-Auswahlfelder zur Verfügung („Blue“ sowie „Red“). Dabei entsprechen die Bezeichnungen der Farbe der nach der Berechnung angezeigten Kurve.

Für die Einbindung eigener Transformatioen siehe A.6.2

Transformations-Methode wählen Über die Drop-Down-Felder (26, 29) kann eine Transformations-Methode ausgewählt werden, die dann auf den im oberen Bereich der A-Scan-Oberfläche geladenen Datensatz durch Betätigen der Buttons „Show“ (28, 31) angewendet werden kann.

Parameter ändern oder beibehalten? Einige Transformationen erwarten vor der Ausführung zusätzliche Parameter. Diese sind in einer gleichnamigen .mat-Datei im Funktionsverzeichnis mit Name und Wert gespeichert. Es kann über die Toggle-Buttons „P“ (27, 30) gewählt werden, ob die bisher verwendeten bzw. vorgegebenen Werte verwendet werden sollen (Toggle-Button deaktiviert), oder ob ein Fenster zur Eingabe und Änderung der Werte angezeigt werden soll (Toggle-Button aktiviert). Standardmäßig sind die Buttons aktiviert.

Vergrößerung anzeigen, als Grafik exportieren Ebenso wie beim A-Scan-Anzeigefeld kann über die Lupen-Schaltfläche (32) ein neues Fenster mit einer Vergrößerung des Transformationsfeldes geöffnet werden. Ein Abspeichern als Grafikdatei ist per Rechtsklick möglich.

Abspeichern der Daten Die Daten der transformierten A-Scans können per Rechtsklick auf das Transformationsfeld (24) > „Save Blue Data as...“ bzw. „Save Red Data as...“ in eine Datei im MATLAB-Datenformat abgespeichert werden.

Auswahlbereich verwenden Transformationen können auch auf einem ausgewählten Bereich aus dem geladenen A-Scan angewendet werden. Hierfür muss zunächst eine Selektion im A-Scan-Anzeigefeld erfolgen (siehe A.3.3). Anschließend kann die Checkbox „Use Region“ (25) aktiviert werden, wodurch anschließend angewandte Transformationen nicht mehr auf dem gesamten Datensatz ausgeführt werden.

A.4 USCT

Die USCT-Oberfläche (Abbildung A.4) dient hauptsächlich der Visualisierung und Auswahl von Sender-Empfänger-Kombinationen.

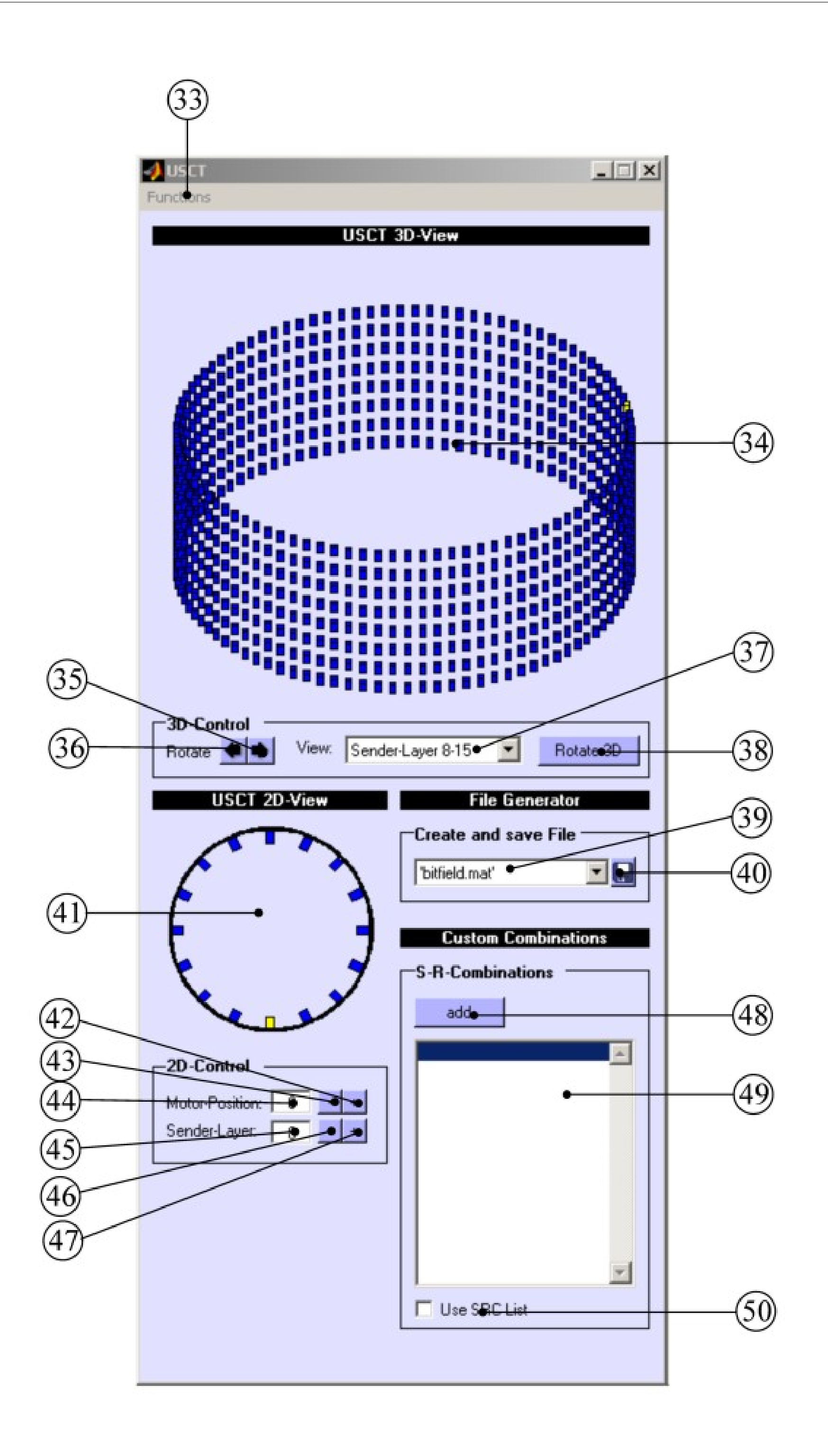


Abbildung A.4: USCT-Oberfläche nach dem Starten der Software

A.4.1 3D-Ansicht

Im oberen Teil der Benutzeroberfläche befindet sich die 3D-Darstellung des USCT (34), wobei jedes dargestellte Element (blau) einem Sender bzw. vier Empfängern entspricht. Ein ausgewählter Sender wird rot markiert, ein ausgewählter Empfänger grün. Dabei ist zu berücksichtigen, dass auch bei nur einem ausgewählten Receiver das ganze Element grün gezeichnet wird.

Ansicht drehen Über die beiden Pfeile „Rotate“ (35, 36) kann die 3D-Ansicht nach links und rechts gedreht werden, um die evtl. hinter anderen versteckten Elemente sichtbar zu machen.

3D-Rotation Der Toggle-Button „Rotate 3D“ (38) ermöglicht im aktivierten Zustand das freie Drehen und Neigen der 3D-Ansicht. Hierfür kann mit der Maus in das Feld der 3D-Ansicht geklickt und mit der gedrückten linken Maustaste und gleichzeitigem Bewegen die Ansicht rotiert werden.

Angezeigte Layer auswählen Über das Drop-Down-Menü „View:“ (37) kann ausgewählt werden, welche Layer im 3D-Modell angezeigt werden sollen. Dabei bieten sich die Optionen den oberen Ring (Sender-Layer 0-7), den mittleren Ring (Sender-Layer 8-15), den unteren Ring (Sender-Layer 16-23) oder den gesamten USCT anzuzeigen. Nach Start der Software wird die Standardeinstellung verwendet, die in den benutzerdefinierten Einstellungen (siehe A.3.1) verändert werden kann.

Sendeelement selektieren Durch einen Rechtsklick auf ein Element öffnet sich ein Kontextmenü, in dem durch „set as sender“ das jeweilige Element als Sender markiert werden kann. Es wird zur Kennzeichnung rot eingefärbt. Eine weitere Möglichkeit ist es, ganze Layer auszuwählen (durch „set layer as“ > „senders“) sowie eine ganze Spalte als Sender zu markieren (durch „set angle as“ > „ „senders“).

Empfangselement selektieren Analog zur Auswahl eines Sendeelements kann mit einem Rechtsklick > „set as receiver“ ein Element als Receiver markiert werden. Da sich hinter einem Element jedoch 4 Receiver verbergen, wird ein Dialog geöffnet, bei dem ein oder mehrere Receiverelemente durch Daraufklicken ausgewählt werden können. Eine Deselektierung innerhalb dieses Dialogs erfolgt ebenfalls durch einen Klick mit der linken Maustaste. Wie auch bei den Sendern ist es möglich komplette Layer oder Winkel auszuwählen.

Elemente deselektieren Über das Kontextmenü, das sich hinter einem Klick mit der rechten Maustaste auf ein Element verbirgt, lassen sich Elemente über den Punkt „deselect“ deselektieren. Hier bietet sich die Möglichkeit die Deselektierung auf einzelne Elemente, alle Elemente, den gesamten Layer oder den gesamten Winkel anzuwenden.

Programmierbare linke Maustaste Zur schnelleren Auswahl von Elementen ist die linke Maustaste programmierbar. Die zuletzt ausgeführte Aktion (z.B einen Layer als Sender definieren, ein einzelnes Element als Receiver selektieren, ein Element deselektieren, ...) kann durch einen Klick mit der linken Maustaste auch auf andere Elemente angewendet werden.

A.4.2 2D-Ansicht

Die 2D-Ansicht (41) stellt jeweils eine Sender-Ebene an einer Motorposition des USCT dar. Dem Benutzer stehen folgende Interaktionsmöglichkeiten zur Verfügung:

Motorposition verändern Im Abschnitt „2D-Control“ kann die Motorposition über den Plus- (42) bzw. Minus-Button (43) verändert werden. Eine explizite Motorposition kann durch Eingabe eines Zahlenwertes zwischen 0 und 5 in das daneben befindliche Textfeld (44) und Drücken der Returntaste angegeben werden.

Sender-Layer verändern Ebenso wie bei der Motorposition kann auch der dargestellte Sender-Layer sowohl über die Buttons (46, 47) als auch über eine direkte Eingabe (45) verändert werden.

Elemente selektieren / deselektieren Das Selektieren bzw. Deselektieren erfolgt analog zur 3D-Ansicht. Durch einen Rechtsklick auf ein Element öffnet sich ein Kontextmenü, das die Unterpunkte „set as Sender“, „set as Receiver“, und „deselect“ anbietet. Bei den Receivern ist zu beachten, dass sich ein neuer Dialog öffnet in dem noch einmal das gewünschte der vier durch den angezeigten Patch repräsentierten Empfangselemente ausgewählt werden kann.

A.4.3 File Generator

Mit dem File-Generator (39, 40) können anhand der markierten Sender- und Empfänger-Elemente Dateien erzeugt werden, die für andere Software benötigt werden:

Bitfield Das Bitfield beschreibt welche Sender- und Empfänger-Kombinationen bei einer Messung vorhanden sind bzw. welche für eine Rekonstruktion eines Bildes verwendet werden. Das erzeugte Ausgabeformat ist eine .mat-Datei mit entsprechenden Datenstrukturen.

AvailableEmitter Available Emitter wird von der Datenakquisitionssoftware „Andromeda“ für die Durchführung einer Messung benötigt. In ihr stehen alle linearen Sendernummern, die bei der Messung verwendet werden sollen. Das Ausgabeformat ist eine .txt-Datei im ASCII Zeichensatz.

AvailableReceiver AvailableReceiver ist das Pendant zur Datei AvailableEmitter. Sie beschreibt, welche Receiverelemente für eine Messung eingesetzt werden.

A.4.4 Beliebige Sender-Empfänger-Kombinationen

Mit dem Listenfeld „Custom Combinations" (49) kann eine Liste beliebiger Sender-Empfänger-Kombinationen angelegt werden.

Hinzufügen einer Sender-Empfänger-Kombination die aktuell markierte(n) Sender-Empfänger-Kombination(en) aus der 3D-Ansicht und den A-Scan-Listenfeldern kann mit dem Button „add" (48) der Liste hinzugefügt werden.

Herauslöschen einer Sender-Empfänger-Kombination Durch Markierung einer oder mehrerer Zeilen im Listenfeld (49) und anschließendem Klick mit der rechten Maustaste öffnet sich ein Kontextmenü, in dem der Unterpunkt „Remove from List" ausgewählt werden kann.

Ganze Liste löschen Die Liste lässt sich durch das Kontextmenü der rechten Maustaste auch vollständig löschen. Hierfür muss der Punkt „Clear List" ausgewählt werden.

Liste verwenden Um die Liste zur Anzeige der jeweiligen A-Scans und für weiterführende Operationen zu benutzen, kann die Checkbox „Use SRC List" (50) aktiviert werden.

A.5 Images

Die Benutzeroberfläche „Images" (Abbildung A.5) arbeitet mit den Daten rekonstruierter Bilder.

A.5.1 Bilder laden und anzeigen

Im Bereich „Image Control" ist es möglich zwei Bilder in das Programm zu laden. Hierfür stehen zwei Buttons mit der Bezeichnung „Load Image" (60, 65) zur Verfügung. Ein Klick darauf öffnet einen Dialog, in dem eine entsprechende Datei ausgewählt werden kann. Zu beachten ist das Datenformat. Verwendet werden rekonstruierte Bilder, die die Bezeichnung „OneImage" enthalten und im MATLAB-Format vorliegen.

Zwischen den Bildern wechseln Sind auf beiden Bild-Kanälen Daten hinterlegt, so kann die Ansicht zwischen ihnen gewechselt werden. Hierfür dienen

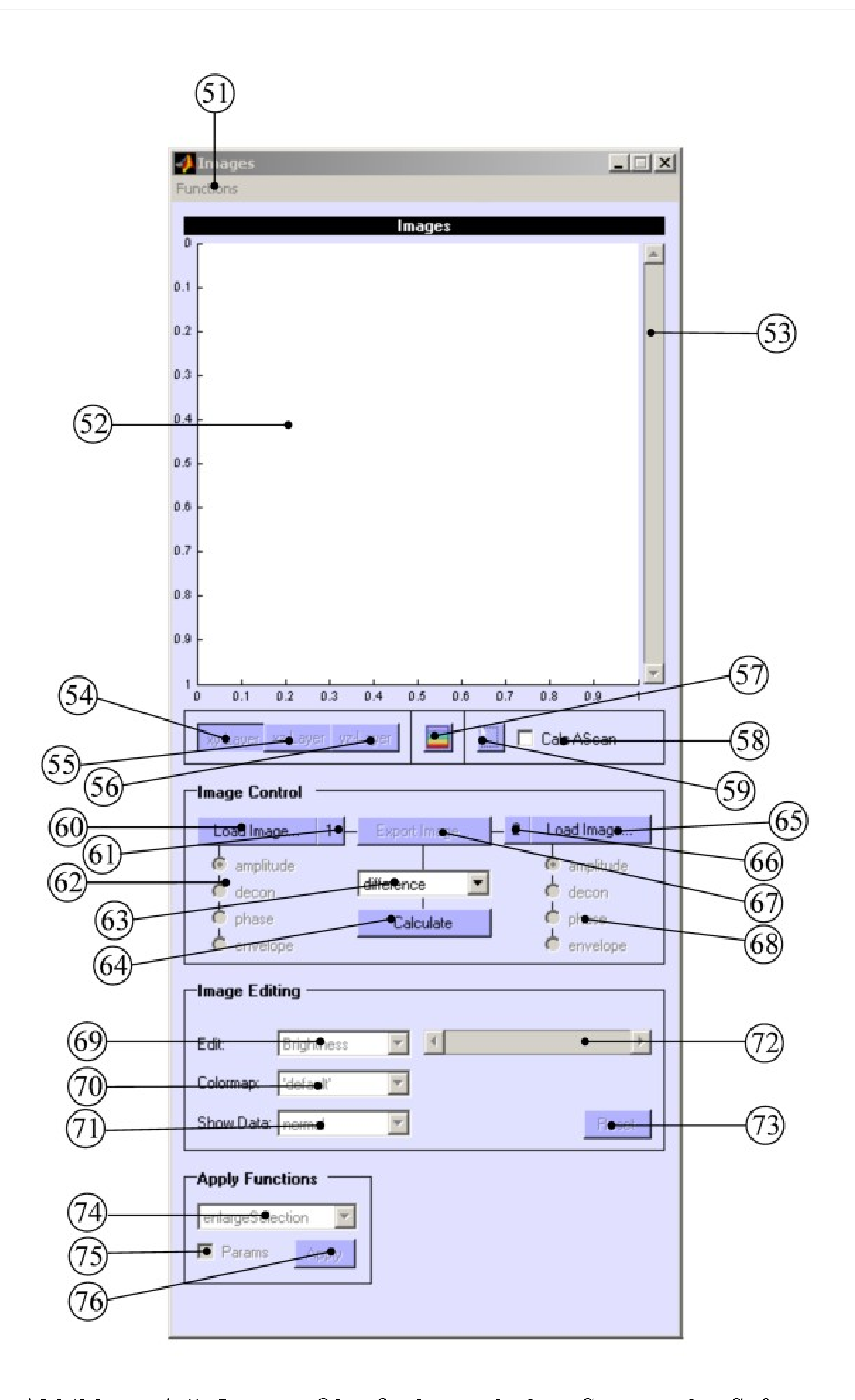


Abbildung A.5: Images-Oberfläche nach dem Starten der Software

die Schaltflächen „1“ und „2“ (61, 66) neben dem jeweiligen „Load-Image“-Button. Alle weiteren Operationen bis auf die Bildbearbeitungsfunktionen

werden dann auf den aktuell angezeigten Daten ausgeführt. Zur Visualisierung wird der zugehörige Button des dargestellten Bildes grün eingefärbt. Ebenso wechselt die Ansicht, wenn auf einer Seite eine Bildart ausgewählt wird.

Bildart auswählen Rekonstruierte USCT-Bilder werden als Datensatz abgespeichert, wobei mehrere Bildarten (Standardwert: das Amplitudenbild, das Phasen-Bild, das Deconvolution-Bild und das Envelope-Bild) gespeichert werden. Zur Auswahl der gewünschten Darstellung besitzt jeder Bildkanal vier Radiobuttons (62, 68), über die das angezeigte Bild bestimmt wird. Die Bildarten sind in der A-Scan-Oberfläche im Menü „Options“ > „Preferences“ anpassbar (siehe A.3.1.

Bilder exportieren Das jeweils angezeigte Bild kann in eine Grafikdatei (.jpg) oder im MATLAB-Datenformat (.mat) exportiert werden. Dies erfolgt über die Schaltfläche „Export Image...“ (67). Im sich öffnenden Speichern-unter-Dialog ist die Angabe des Dateinamens sowie der Dateiendung (!) zu beachten, da anhand dieser das Ausgabeformat gewählt wird.

JPEG-Bilder werden in einer Qualität von 100 exportiert und berücksichtigen die vorgenommenen Bildbearbeitungen.

Exportierte Daten im MATLAB-Datenformat erhalten eine Variable die nach der ausgewählten Bildart benannt ist.

Informationen zum Bild anzeigen Als Tooltip der beiden Buttons „1“ und „2“ (61, 66) werden Informationen zu dem jeweiligen geladenen Bild angezeigt. Zur Anzeige des Tooltips muss der Mauszeiger kurze Zeit über dem Button stillgehalten werden.

A.5.2 Volumenbild-Funktionen

Die Images-Oberfläche ermöglicht auch das Laden von Volumenbildern. Dargestellt wird jeweils ein Schichtbild daraus. Eine Anzeige in der oberen rechten Ecke des Bildes zeigt dabei den Layer des derzeitigen Bildes an.

Layer wechseln Durch Verschieben der Bildlaufleiste (53) rechts des Bildes wird ein neuer Layer ausgewählt und das entsprechende Schichtbild angezeigt. Somit ist es möglich das komplette Volumen zu durchlaufen.

Orientierung ändern über die Buttons „xy-Layer“ (54), „xz-Layer“ (55) und „yz-Layer“ (56) kann die Orientierung des Durchlaufens verändert werden. Ein gedrückter Tooglebutton „xy-Layer“ beispielsweise erzeugt Schichtbilder aus der xy-Ebene, d.h. aus der Sicht von oben auf den USCT-Zylinder. xz-Layer und yz-Layer ändern dies entsprechend in Schichtbilder aus der Frontalansicht bzw. Seitenansicht.

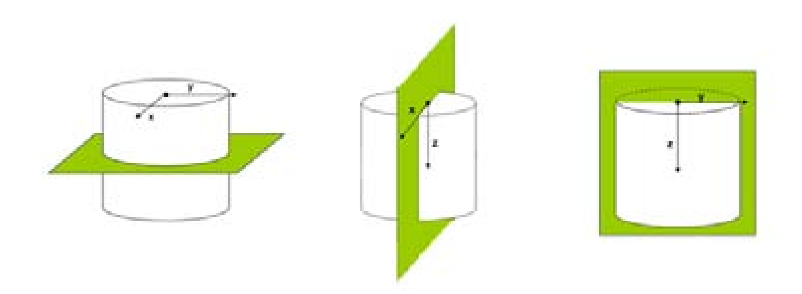


Abbildung A.6: Schichtbilder bei einem Volumen

A.5.3 Bildvereinigungen

Aus zwei geladenen Bildern können Vereinigungen wie z.B. das Differenzbild, eine Summation der beiden Bilder, usw. gebildet werden. Hierfür kann eine Vereinigungsfunktion im Drop-Down-Menü (63) ausgewählt und anschließend mit dem Klicken des „Calculate“-Buttons (64) ausgeführt werden. Zur Veranschaulichung welches Bild angezeigt wird erhält der Button eine grüne Farbe.

A.5.4 Bildbearbeitungsfunktionen

Im Bereich „Image Editing“ bietet sich die Möglichkeit das Bild mit einigen Funktionen zu bearbeiten:

Intensität verändern Es kann über das Drop-Down-Feld (69) die Änderung der Helligkeit ausgewählt werden. Mit der Bildlaufleiste (72) rechts daneben kann dann der Helligkeitswert angepasst werden.

Colormap verändern Da die geladenen Rekonstruktionen von MATLAB als indexierte Bilder dargestellt werden, kann die Colormap (d.h. eine Lookup-Tabelle, bei der jedem Zahlenwert des Bildes eine Farbe zugeordnet ist) verändert werden. Zur Auswahl stehen im zugehörigen Drop-Down-Menü (70) die von MATLAB vorgegebenen Colormaps.

Daten logarithmisch anzeigen Über „Show Data“ (71) kann ausgewählt werden, ob die Intensitätswerte des Bildes normal oder logarithmisch angezeigt werden.

A.5.5 Bereichsauswahl

Mit dem Auswahl-Werkzeug (59) kann ein Bereich im Bild markiert werden. Auf ihn können nun weitere Funktionen angewendet werden (siehe A.5.6). Gleichzeitig wird die Auswahl im 3D-Modell des Ultraschall-Computer-Tomographen in der USCT-Benutzeroberfläche dargestellt und je nach Auswahl der Checkbox „Calc

A-Scan“ (58) zum entsprechenden Bereich im A-Scan zurückgerechnet. Damit wird ersichtlich aus welchen Daten sich die Auswahl zusammensetzt.

Zu beachten ist, dass das rekonstruierte Bild mit dem ausgewählten Experiment übereinstimmen sollte, da nur so eine Zuordnung erfolgen kann.

A.5.6 Funktionen und Berechnungen auf Selektionen

Auf der Bereichsauswahl können zusätzlich beliebige Funktionen ausgeführt werden. Die verfügbaren Funktionen stehen im Drop-Down-Menü (74) unter „Apply Functions“ zur Auswahl. Durch einen Klick auf „Apply“ (76) wird die ausgewählte Funktion ausgeführt.

Durch Aktivierung der Checkbox „Params“ (75) können die Eingabeparameter verändert werden. Es wird bei der Ausführung der Funktion ein Dialog geöffnet, in dem die aktuellen Werte eingesehen und verändert werden können.

Es können beliebige neue Funktionen hinzugefügt werden. Bei der Programmierung dieser sind allerdings die Schnittstellenspezifikationen zu beachten. Hierfür siehe A.6.4.

A.6 Erweiterung des DiagTools

A.6.1 Unabhängige Funktionen in den Menüs

Jeder der drei Teile der Benutzeroberfläche verfügt in der Menüleiste über das Menü „Functions“, in dem vom Rest des Programms abhängige oder unabhängige Funktionen eingehängt werden können. Diese werden durch ein Anklicken im Menü aufgerufen. Beim Einhängen von Funktionen ist zu beachten:

- Die implementierte Funktion muss in einem gleichnamigen .m-File im Unterordner ~/functions/usctGUIMenu , ~/functions/imagesGUIMenu bzw. ~/functions/ascanGUIMenu gespeichert werden.

- Eine gleichnamige .mat-Datei muss im gleichen Verzeichnis liegen und muss die Variable „label“ enthalten, die den Titel der Funktion angibt, der im Menü des DiagTools erscheint.

- Zusätzlich ist es denkbar weitere Parameter, Daten, etc. in dieser Datei zu speichern, die die Funktion für die Ausführung benötigt.

Bei Initialisierung des DiagTools werden die Ordner ausgelesen und die Einträge dem Menü „Functions“ als Unterpunkte hinzugefügt. Was die Funktion berechnet, anzeigt etc. kann der Benutzer frei bestimmen, z.B. auch die GUI - hier wird vom DiagTool nichts automatisch erzeugt! Als Beispiel kann die Implementierung „Visualize Geometry Files“ angesehen werden, die sich im Ordner ~/functions/usctGUIMenu befindet.

A.6.2 A-Scan Transformationen

Um dem DiagTool eine neue A-Scan-Transformation hinzuzufügen ist folgendes zu beachten:

Implementierung der Transformation Zunächst muss die Transformations-Funktion implementiert werden:

- Der Funktionsname der Transformation entspricht dem Dateinamen der .m-Datei, in dem diese abgelegt wurde.
- Die Transformation erhält als ersten Parameter immer die Daten des aktuell im DiagTool ausgewählten A-Scans.
- Der zweite Parameter „inputParams“ ist optional. Benötigt die Tranformation weitere Eingabewerte, so muss die implementierte Funktion diesen Parameter erhalten.
- Die Transformations-Funktion muss ein Datenarray zurückliefern, welches im DiagTool anschließend geplottet wird.

inputParams anlegen Verwendet die implementierte Funktion zusätzliche Parameter, muss eine Datei mit der Variablen inputParams angelegt werden:

- Die Variable inputParams ist ein MATLAB-Struct, in dem beliebige Parameter gespeichert sein können.
- Werden für die Transformation außer dem A-Scan weitere Parameter benötigt, so müssen diese in der Variable „inputParams“ in einem .mat -File mit gleichem Namen wie das .m-File der Transformation abgespeichert werden.

Weiteres Außerdem ist zu beachten:

- Das .m-File sowie (falls verwendet) das .mat-File müssen im Unterordner ~/functions/ascans/ abgelegt werden.
- DiagTool liest beim Starten den Ordner aus und bietet die Funktionen in den Drop-Down-Menüs an.
- Bei Aufruf der Funktion wird überprüft, ob das .mat-File vorhanden ist. Falls ja wird - wenn gewünscht - ein Dialog zur Eingabe/Änderung der Parameter angezeigt und durch Bestätigen mit OK die Funktion ausgeführt sowie die berechneten Daten geplottet.
- Ist die .mat-Datei nicht vorhanden, wird überprüft, ob die Funktion weitere Parameter erwartet. Ist dies der Fall erscheint eine Fehlermeldung, andernfalls wird die Funktion ausgeführt und die berechneten Daten geplottet.

Beispiel: DetectPulses.m beginnt mit

```
function [peakAscan] = DetectPulses(ascan, inputParams)
```

Beispiel: DetectPulses.mat enthält die Variable „inputParams“

```
inputParams =

         inputPulse: [80x1 double]
    inputPulseBegin: 1
      inputPulseEnd: 80
           startPos: 1
               step: 10
       numDftPoints: 256
          frequency: 10000000
              limit: 1.0000e-010
              scale: 1
```

A.6.3 Bildvereinigungs-Funktionen

Die „Images“-Oberfläche erlaubt es Bildvereinigungs-Funktionen der beiden geladenen Bilder auszuführen. Bei der Implementierung solcher ist zu beachten:

- Der Funktionsname muss dem Namen der Datei entsprechen, in dem sie abgelegt ist.
- Als Parameter erhält die Funktion die beiden kompletten Datensätze (z.B. bei einem Volumenbild ein dreidimensionales Array) der geladenen Bilder.
- die Funktion muss als Rückgabewert ein zwei- bzw. dreidimensionales Array liefern, das als Bild (bzw. Volumen) dargestellt werden kann.
- eine Fehlerbehandlung muss innerhalb der implementierten Funktion erfolgen. So z.B. wenn Operationen nur auf gleich großen Matrizen möglich sind.
- die Datei ist im Verzeichnis ~/functions/imageComparison abzulegen.
- beim Starten der Software wird der Ordner ausgelesen und das Drop-Down-Menü mit den Namen der Funktionen gefüllt.

Das Interface stellt sich in Code-Form wie folgt dar:

```
function newImage = <funktionsname>(imageData1, imageData2)
```

A.6.4 Allgemeine Bild-Funktionen

Eine zusätzliche Erweiterungsmöglichkeit hat die Software mit den allgemeinen Bild-Funktionen, die auf dem gesamten Bild oder einer Auswahl (Region of Interest, ROI) arbeiten. Hierfür wurde eine Schnittstelle geschaffen, bei der die folgenden Punkte zu beachten sind:

- Der Name der Funktion muss dem Namen der Datei entsprechen in der sie gespeichert ist.

- Als Parameter erhält die Funktion zunächst die aktuellen Bilddaten. Als Standard sind dies die 2D-Daten des dargestellten Layers. Über einen Eintrag in einer Parameterdatei lässt sich dies auch auf die Übergabe des 3D-Datensatzes abändern.

- Zweiter Parameter ist die Nummer des Layers, der nur bei einem 3D-Datensatz relevant ist. Bei 2D-Daten wird der Wert 1 übergeben.

- Als dritten Parameter erhält die Funktion die Koordinaten der Auswahl aus dem Bild.

  `selection(1,1)` liefert dabei die Entfernung der linken oberen Auswahl-Ecke vom linken Bildrand.

  `selection(1,2)` liefert den Abstand der linken oberen Auswahl-Ecke vom oberen Bildrand.

  `selection(2,1)` liefert die Entfernung der rechten unteren Ecke der Auswahl vom linken Bildrand.

  `selection(2,2)` liefert den Abstand der rechten unteren Ecke vom oberen Bildrand.

  Zu beachten ist das Auswählen der Elemente entsprechend der markierten Region aus der Bild-Matrix. Es werden zunächst die y-Koordinaten der beiden Selektionspunkte benötigt, um die Zeilen auszuwählen, die x-Koordinaten dann für die Spalten.

- Alternativ erhält die Funktion als vierten Parameter die Variable „inputParams“, die wie schon bei den A-Scan-Transformationen in einer .mat-Datei gleichen Namens wie die Funktion gespeichert ist und im selben Verzeichnis liegen muss. Spezielle Bedeutung hat dabei der Parameter inputParams.dataMode. Mit diesem kann ausgewählt werden, ob eine Funktion lediglich die 2D-Daten des aktuell dargestellten Bildes erhält (Wert ’2D’) oder ob es den kompletten 3D-Datensatz übergeben bekommt (Wert ’3D’).

- Die Funktion besitzt keinen Rückgabewert.

- Die Anzeige der Ergebnisse einer Berechnung sind dem Programmierer selbst überlassen, hierfür stellt das DiagTool keine gesonderte Funktionalität zur Verfügung!
- Auch die Fehlerbehandlung sowie die Versionskompatibilität unterliegt dem Programmierer.

A.6.5 Vorlagen

Programmier-Vorlagen für die Erweiterung der A-Scan-Transformationen, der Bildvereinigungs-Funktionen und der Bild-Funktionen finden sich im Unterordner ~/templates. Zu beachten ist bei Verwendung der Vorlagen, dass die Funktion innerhalb der Datei umbenannt und die Datei selbst unter entsprechendem Namen im zugehörigen Verzeichnis abgespeichert wird!

Anhang B

Dateiformate

B.1 AvailableEmitters.txt

```
# Available Emitters
# Simple textfile, whitespace separated list of numbers.
# Numbering is according to counterclockwise enumeration of emitters
# WITHOUT concerning about motor positions.
# # denotes single line comment
#
#
# First: 0
# Last : 383
#
128 132 136 140
```

B.2 AvailableReceivers.txt

```
# Available Receivers
# Simple textfile, whitespace separated list of numbers.
# Numbering is according to counterclockwise enumeration of receivers
# WITHOUT concerning about motor positions.
# # denotes single line comment
#
# First: 0
# Last : 1535
#
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
```

B.3 codedExcitation.txt

```
# 510 * 16 bit werte fuer die coded excitation
# (und davon nur 10bit genutzt)
#
512
522
536
549
556
553
538
512
478
446
421
414
427
462
512
567
616
648
652
627
577
512
444
389
359
```

Anhang C

Struktur der Software

Das folgende Schaubild (Abbildung C.1) zeigt die Interaktion der primären Funktionen, die in der Software benutzt werden. Darüber hinaus gibt es innerhalb der genannten Dateien meist noch eine ebenso große Interaktion zwischen den beinhalteten Subfunktionen. Die gelb bzw. orange gezeichneten Felder stellen die Schnittstellen dar, die eine unbegrenzte Anzahl weiterer Funktionen enthalten können. Die grünen Felder visualisieren die drei Hauptoberflächen sowie die Funktion zum Starten der Software „DiagToolStarter“. Rot dargestellt ist die zentrale Datei für die Speicherung von Konstanten, „constants.m“. Nicht abgebildet sind zusätzlich verwendete MATLAB-Datendateien.

Über die Art und Zeitpunkte der gegenseitigen Aufrufe informiert die mit der Software gelieferte Funktionsreferenz im Unterordner „doc“. Dort werden im Unterpunkt der Referenzseite einer Funktion A mit dem Titel „This function calls“ Funktionen B,C,D,... aufgelistet, die von Funktion A aufgerufen werden. Umgekehrt sind unter „This function is called by“ Funktionen B,C,D,... aufgelistet, die Funktion A aufrufen. Des Weiteren können in den Referenzseiten die Subfunktionen eingesehen werden.

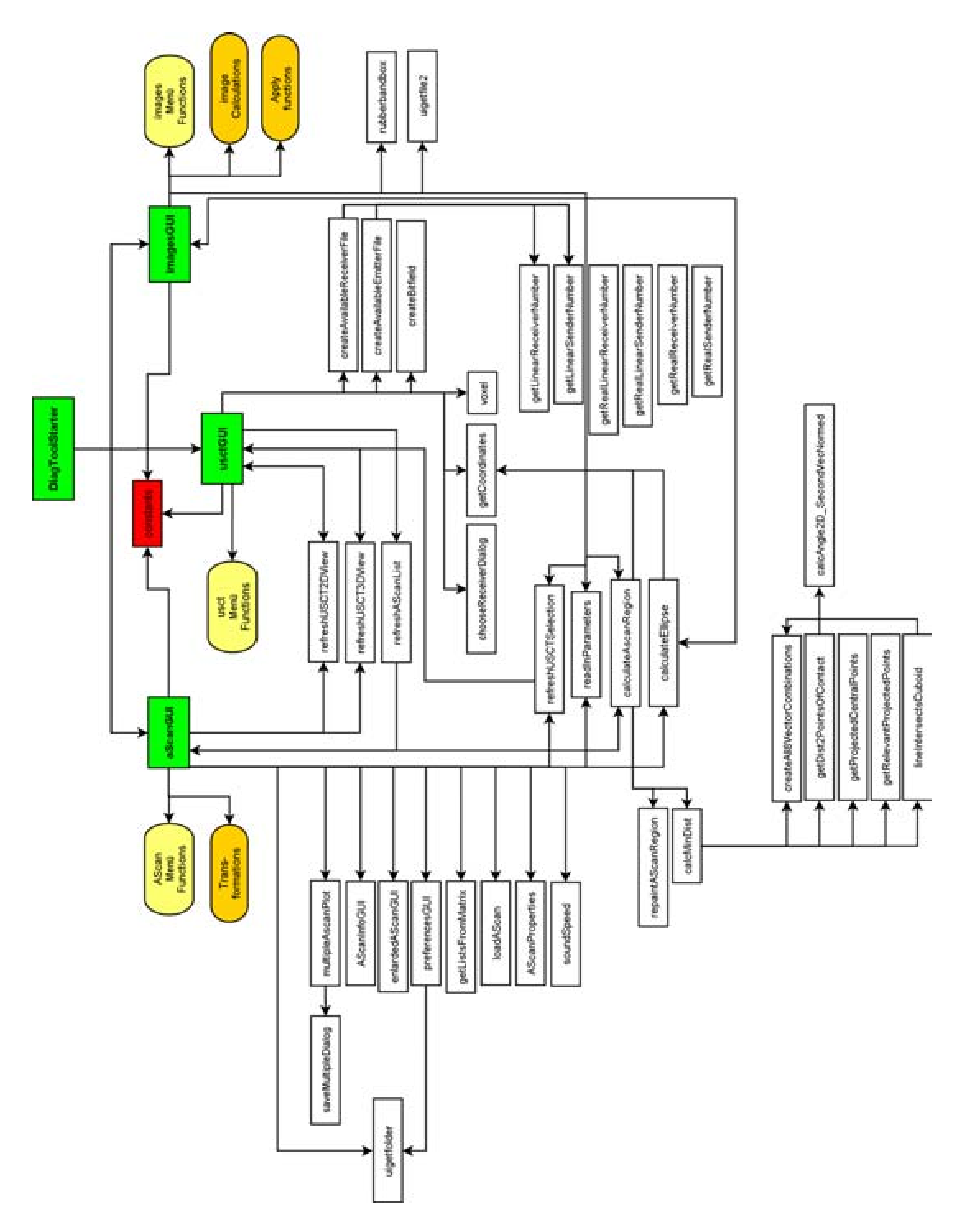


Abbildung C.1: Struktur der Software

Anhang D

Inhalt der CD

Auf der beiliegenden CD befinden sich folgenden Daten in den angegebenen Unterordnern:

Software

- **Gepackt:** Die erstellte Software in einem gepackten zip-Archiv. Unter anderem in dieser Form soll die Verteilung der Software erfolgen.
- **Sources:** Die entpackte Software mit der beschriebenen Ordnerstruktur. Die .m-Dateien beinhalten die Quelltexte und können sowohl mit dem MATLAB-Editor als auch mit jedem anderen beliebigen Text-Editor geöffnet werden.
- **Benutzerhandbuch:** Das Benutzerhandbuch zur Software als PDF-Dokument und Online-Version in Form von HTML-Dokumenten

Quellen

- Hier finden sich die im Literaturverzeichnis angegebenen Quellen, welche als PDF-Dokumente vorliegen. Anhand des Kürzels in eckigen Klammern des Dateinamens kann eine Zuordnung zum Literaturverzeichnis erfolgen. Verwendete Bücher, die nicht im PDF-Format vorliegen, sind darin nicht enthalten.

Diplomarbeit

- **PDF:** Die vorliegende Diplomarbeit als PDF-Dokument
- **HTML:** Die Diplomarbeit als Online-Version im HTML-Format